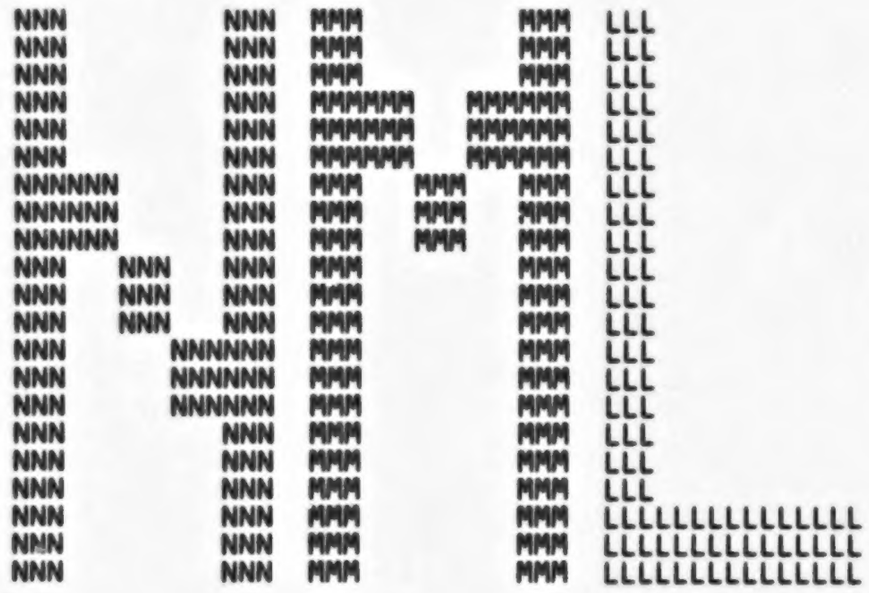
NNN NNN MMM MMM LLL
NNN NNN MMM MMM LLL
NNN NNN MMM MMM LLL
NNN NNN MMMMMM MMMMMM LLL
NNN NNN MMMMMM MMMMMM LLL
NNN NNN MMMMMM MMMMMM LLL
NNNNNN NNN MMM MMM MMM LLL
NNNNNN NNN MMM MMM MMM LLL
NNNNNN NNN MMM MMM MMM LLL
NNN NNN NNN MMM MMM LLL
NNN NNN NNN MMM MMM LLL
NNN NNN MMM MMM LLL
NNN NNNNNN MMM MMM LLL
NNN NNNNNN MMM MMM LLL
NNN NNNNNN MMM MMM LLL
NNN NNN MMM MMM LLL
NNN NNN MMM MMM LLL
NNN NNN MMM MMM LLL
NNN NNN MMM MMM LLLLLLLLLLLLLLL
NNN NNN MMM MMM LLLLLLLLLLLLLLL
NNN NNN MMM MMM LLLLLLLLLLLLLLL

FILEID**NMLNODFIL

```
NN      NN  MM      MM  LL          NN      NN    000000    DDDDDDDD    FFFFFFFFFF    IIIIII    LL
NN      NN  MM      MM  LL          NN      NN    000000    DDDDDDDD    FFFFFFFFFF    IIIIII    LL
NN      NN  MMMM  MMMM  LL          NN      NN  00      00  DD      DD  FF              II      LL
NN      NN  MMMM  MMMM  LL          NN      NN  00      00  DD      DD  FF              II      LL
NNNN    NN  MM  MM  MM  LL          NNNN    NN  00      00  DD      DD  FF              II      LL
NNNN    NN  MM  MM  MM  LL          NNNN    NN  00      00  DD      DD  FF              II      LL
NN  NN  NN  MM      MM  LL          NN  NN  NN  00      00  DD      DD  FFFFFFFF        II      LL
NN  NN  NN  MM      MM  LL          NN  NN  NN  00      00  DD      DD  FFFFFFFF        II      LL
NN    NNNN  MM      MM  LL          NN    NNNN  00      00  DD      DD  FF              II      LL
NN    NNNN  MM      MM  LL          NN    NNNN  00      00  DD      DD  FF              II      LL
NN      NN  MM      MM  LL          NN      NN  00      00  DD      DD  FF              II      LL                  ....
NN      NN  MM      MM  LL          NN      NN  00      00  DD      DD  FF              II      LL                  ....
NN      NN  MM      MM  LLLLLLLLLL  NN      NN    000000    DDDDDDDD    FF            IIIIII  LLLLLLLLLL          ....
NN      NN  MM      MM  LLLLLLLLLL  NN      NN    000000    DDDDDDDD    FF            IIIIII  LLLLLLLLLL          ....

LL              IIIIII    SSSSSSSS
LL              IIIIII    SSSSSSSS
LL                II    SS
LL                II    SS
LL                II    SS
LL                II    SS
LL                II      SSSSSS
LL                II      SSSSSS
LL                II            SS
LL                II            SS
LL                II            SS
LL                II            SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;  1          0001  0 %TITLE  'Node File Routines for Network Management'
;  2          0002  0 MODULE NMLNODFIL (
;  3          0003  0                   LANGUAGE (BLISS32),
;  4          0004  0                   ADDRESSING_MODE (NONEXTERNAL=GENERAL),
;  5          0005  0                   ADDRESSING_MODE (EXTERNAL=GENERAL),
;  6          0006  0                   IDENT = 'V04-000'
;  7          0007  0                   ) =
;  8          0008  1 BEGIN
;  9          0009  1
;  10         0010  1 !
;  11         0011  1 !*****************************************************************************
;  12         0012  1 !*                                                                           *
;  13         0013  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;  14         0014  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;  15         0015  1 !*  ALL RIGHTS RESERVED.                                                      *
;  16         0016  1 !*                                                                           *
;  17         0017  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;  18         0018  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;  19         0019  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;  20         0020  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;  21         0021  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;  22         0022  1 !*  TRANSFERRED.                                                              *
;  23         0023  1 !*                                                                           *
;  24         0024  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;  25         0025  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;  26         0026  1 !*  CORPORATION.                                                              *
;  27         0027  1 !*                                                                           *
;  28         0028  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;  29         0029  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;  30         0030  1 !*                                                                           *
;  31         0031  1 !*                                                                           *
;  32         0032  1 !*****************************************************************************
;  33         0033  1 !
;  34         0034  1
;  35         0035  1 !++
;  36         0036  1 ! FACILITY:     DECnet Network Management Listener (NML)
;  37         0037  1 !
;  38         0038  1 ! ABSTRACT:
;  39         0039  1 !
;  40         0040  1 !       This module contains routines which manage the node permanent database
;  41         0041  1 !       files used by network management.  This file contains permanent data
;  42         0042  1 !       about the configuration of nodes in the network.
;  43         0043  1 !
;  44         0044  1 !       When AREA support was added to DECnet, the node database grew to a
;  45         0045  1 !       size that made the old database too slow (a SHOW NODE FOO searched
;  46         0046  1 !       through the database reading one record at a time, until FOO was
;  47         0047  1 !       found.  This module was created to use a four keyed file that
;  48         0048  1 !       allows single $GETs and $PUTs for each node, which is much faster.
;  49         0049  1 !       All other entities permanent databases have been left in the old
;  50         0050  1 !       format.
;  51         0051  1 !
;  52         0052  1 ! ENVIRONMENT:  VAX/VMS Operating System
;  53         0053  1 !
;  54         0054  1 ! AUTHOR:       Kathy Perko     , CREATION DATE:  6-July-1983
;  55         0055  1 !
;  56         0056  1 ! MODIFIED BY:
;  57         0057  1 !       V03-005 MKP0005         Kathy Perko             2-July-1984
```

```
;    58          0058  1 !                 Fix previous fix so that PURGE KNOWN NODES works.  The
;    59          0059  1 !                 RFA is cleared when a record is deleted, so the check
;    60          0060  1 !                 to see if the RFA has changed between passes, and the subsequent
;    61          0061  1 !                 $GET gets an EOF.  Enhance the RFA check to skip the
;    62          0062  1 !                 $GET if the RFA is zero.
;    63          0063  1 !
;    64          0064  1 !           V03-004 MKP0004         Kathy Perko             23-April-1984
;    65          0065  1 !                 Change NML$READ_KNOWN_NODE_REC to save the RFA (record file
;    66          0066  1 !                 address) in case an intermediate operation (between the
;    67          0067  1 !                 sequential reads) moves the "next record".  If the RFA
;    68          0068  1 !                 has changed, do a $GET before reading the next node
;    69          0069  1 !                 record.
;    70          0070  1 !
;    71          0071  1 !           V03-003 MKP0003         Kathy Perko             31-Mar-1984
;    72          0072  1 !                 Move the node database conversion to the upgrade module
;    73          0073  1 !                 (NMLUPGRAD)
;    74          0074  1 !
;    75          0075  1 !           V03-002 MKP0002         Kathy Perko             2-Mar-1984
;    76          0076  1 !                 Fix node file create to use default name string of
;    77          0077  1 !                 SYS$SYSTEM:.DAT.
;    78          0078  1 !
;    79          0079  1 !           V03-001 MKP0001         Kathy Perko             7-Feb-1984
;    80          0080  1 !                 When converting the node database from the old format
;    81          0081  1 !                 to the new, do the conversion to a temporary file in case
;    82          0082  1 !                 the system crashes part way through.  Rename the file to
;    83          0083  1 !                 it's correct name when done.
;    84          0084  1 !
;    85          0085  1 !--
```

```
;    87          0086  1 %SBTTL  'Definitions'
;    88          0087  1
;    89          0088  1 !
;    90          0089  1 ! TABLE OF CONTENTS:
;    91          0090  1 !
;    92          0091  1
;    93          0092  1 FORWARD ROUTINE
;    94          0093  1     NML$OPEN_NODE_FILE,                 ! Open node database file
;    95          0094  1     NML$CLOSE_NODE_FILE,                ! Close node database file
;    96          0095  1     NML$READ_NODE_REC,                  ! Read a record from node database file
;    97          0096  1     NML$WRITE_NODE_REC,                 ! Write a record from node database file
;    98          0097  1     NML$DELETE_NODE_REC,                ! Delete a record from node database file
;    99          0098  1     NML_MAP_KEYS,                       ! Set up key used to access node db.
;   100          0099  1     NML$READ_LOOPNODE,                  ! Get a loopnode for a specified circuit
;   101          0100  1     NML$READ_KNOWN_NODE_REC,            ! Read known records from node
;   102          0101  1                                         !       database file
;   103          0102  1     NML$CREATE_NODE_DB,                 ! Create new node database file
;   104          0103  1     NML$CONNECT_NODE_RAB;               ! Connect RAB for node database file
;   105          0104  1 !
;   106          0105  1 ! INCLUDE FILES:
;   107          0106  1 !
;   108          0107  1
;   109          0108  1 LIBRARY 'LIB$:NMLLIB.L32';
;   110          0109  1 LIBRARY 'SHRLIB$:NMALIBRY.L32';
;   111          0110  1 LIBRARY 'SYS$LIBRARY:STARLET.L32';
;   112          0111  1
;   113          0112  1 !
;   114          0113  1 ! OWN STORAGE:
;   115          0114  1 !
;   116          0115  1 OWN
;   117          0116  1     nml$a_netnode_fab:          $FAB_DECL,
;   118          0117  1     nml$a_netnode_rab:          $RAB_DECL,
;   119          0118  1     nml$a_protection_xab:       $XABPRO_DECL,
;   120          0119  1     nml$a_summary_xab:          $XABSUM_DECL,
;   121          0120  1     nml$a_node_address_xab:     $XABKEY_DECL,
;   122          0121  1     nml$a_node_name_xab:        $XABKEY_DECL,
;   123          0122  1     nml$a_node_type_xab:        $XABKEY_DECL,
;   124          0123  1     nml$a_node_list_xab:        $XABKEY_DECL,
;   125          0124  1     nml$t_key_value:            VECTOR [3, WORD];
;   126          0125  1
;   127          0126  1 GLOBAL
;   128          0127  1     nml$gq_node_file_dsc        : VECTOR [2]
;   129          0128  1                 INITIAL (%CHARCOUNT ('NETNODE'),
;   130          0129  1                          UPLIT BYTE ('NETNODE'));
;   131          0130  1
;   132          0131  1 EXTERNAL LITERAL
;   133          0132  1     nml$_nodcvterr;
;   134          0133  1
;   135          0134  1 !
;   136          0135  1 ! Declare common NML external references.
;   137          0136  1 !
;   138          0137  1 $nml_extdef;
;   139          0138  1
;   140          0139  1 EXTERNAL ROUTINE
;   141          0140  1     nma$searchfld,
;   142          0141  1     nml$chkfileio,
;   143          0142  1     nml$upgrade_perm_dbs,
```

```
;  144     0143  1     nml$debug_txt,
;  145     0144  1     nml$logfileop,
;  146     0145  1     nml$logrecordop;
;  147     0146  1
```

```
;  149          0147  1 %SBTTL 'nml$open_node_file  Open node permanent database file'
;  150          0148  1 GLOBAL ROUTINE nml$open_node_file =
;  151          0149  1
;  152          0150  1 !++
;  153          0151  1 ! FUNCTIONAL DESCRIPTION:
;  154          0152  1 !       This routine opens the node permanant database file.
;  155          0153  1 !
;  156          0154  1 ! FORMAL PARAMETERS:
;  157          0155  1 !       None
;  158          0156  1 !
;  159          0157  1 ! ROUTINE VALUE:
;  160          0158  1 ! COMPLETION CODES:
;  161          0159  1 !       Failure or RMS error
;  162          0160  1 !
;  163          0161  1 !--
;  164          0162  1
;  165          0163  2 BEGIN
;  166          0164  2
;  167          0165  2 LOCAL
;  168          0166  2     fab:        REF BBLOCK,
;  169          0167  2     status;
;  170          0168  2
;  171          0169  2 status = rms$_suc;
;  172          0170  2 fab = nml$a_netnode_fab;
;  173          0171  2 IF .fab [fab$w_ifi] EQL 0 THEN          ! If file isn't open, do it.
;  174          0172  3     BEGIN
;  175          0173  3     !
;  176          0174  3     ! Open node database.  If there isn't one, create a new node database
;  177          0175  3     ! file.  If the open succeeds, but the file only has one key, it's the
;  178          0176  3     ! old node database format, so convert it.
;  179          0177  3     !
;  180        P 0178  3     $FAB_INIT ( FAB = .fab,
;  181        P 0179  3                 FAC = (GET,PUT,UPD,DEL),
;  182        P 0180  3                 DNM = 'SYS$SYSTEM:.DAT',        ! Default filename string
;  183        P 0181  3                 FNA = .nml$gq_node_file_dsc [1],
;  184        P 0182  3                 FNS = .nml$gq_node_file_dsc [0],
;  185        P 0183  3                 SHR = (UPD, PUT, GET, DEL),     ! File sharing options
;  186        P 0184  3                 XAB = nml$a_summary_xab         ! XAB Chain
;  187          0185  3                 );
;  188          0186  3     $XABSUM_INIT (XAB = nml$a_summary_xab);     ! XAB address
;  189          0187  3
;  190          0188  3     status = $OPEN (FAB = .fab);
;  191          0189  3     IF .status THEN
;  192          0190  4         BEGIN
;  193          0191  4         !
;  194          0192  4         ! If the node file has one key, it's the old node database format,
;  195          0193  4         ! so do a conversion to the new format.
;  196          0194  4         !
;  197          0195  4         IF .nml$a_summary_xab [xab$b_nok] EQL 1 THEN
;  198          0196  5             BEGIN
;  199          0197  5             !
;  200          0198  5             ! Close old permanent database (which was opened using the
;  201          0199  5             ! new permanent database XABs, etc.).
;  202          0200  5             !
;  203          0201  5             nml$close_node_file ();
;  204          0202  5             !
;  205          0203  5             ! Do a V4.0 upgrade on the permanent database files.  The upgrade
```

```
;  206          0204  5              ! procedure will force this call.  The procedure involves converting
;  207          0205  5              ! area 0 to either a customer supplied area number or area 1, and
;  208          0206  5              ! it involves converting the node database to a faster format.
;  209          0207  5              !
;  210          0208  5              status = nml$upgrade_perm_dbs ();
;  211          0209  5              IF .status THEN
;  212          0210  6                  BEGIN
;  213          0211  6                  nml$a_netnode_fab [fab$l_fna] = .nml$gq_node_file_dsc [1];
;  214          0212  6                  nml$a_netnode_fab [fab$b_fns] = .nml$gq_node_file_dsc [0];
;  215          0213  6                  status = $OPEN (FAB = .fab);
;  216          0214  5                  END;
;  217          0215  4              END;
;  218          0216  4          END
;  219          0217  3      ELSE
;  220          0218  3          !
;  221          0219  3          ! If the node database doesn't already exist, create one and
;  222          0220  3          ! connect the RAB record stream.
;  223          0221  3          !
;  224          0222  3          IF .status EQL rms$_fnf THEN
;  225          0223  3              status = nml$create_node_db (nml$gq_node_file_dsc, fab);
;  226          0224  3      !
;  227          0225  3      ! Connect the RAB to the file.
;  228          0226  3      ! If NML$LOG is defined with file io bit set, log a "file opened"
;  229          0227  3      ! message.
;  230          0228  3      !
;  231          0229  3      IF .status THEN
;  232          0230  4          BEGIN
;  233          0231  4          status = nml$connect_node_rab ();
;  234          0232  4          nml$logfileop (dbg$c_fileio,
;  235          0233  4                         nma$c_opn_node,
;  236          0234  4                         $ASCID ('file opened.'));
;  237          0235  3          END;
;  238          0236  2      END;
;  239          0237  2  RETURN .status;
;  240          0238  2
;  241          0239  1  END;                ! of NML$OPEN_NODE_FILE


                                                                .TITLE  NMLNODFIL Node File Routines for Network Manage
                                                                        ment
;                                                               .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                00000 P.AAA:    .ASCII  \NETNODE\                                        ;
                                    45  44  4F  4E  54  45  4E  00007 P.AAB:    .ASCII  \SYS$SYSTEM:.DAT\                                ;
54  41  44  2E  3A  4D  45  54  53  59  53  24  53  59  53  00016 P.AAD:    .ASCII  \file opened.\                                   ;
        2E  64  65  6E  65  70  6F  20  65  6C  69  66  00022            .BLKB   2
                                            0000000C    00024 P.AAC:    .LONG   12                                               ;
                                            00000000'   00028            .ADDRESS P.AAD                                           ;

                                                                .PSECT  $GWN$,NOEXE,2

                                                00000 NML$A_NETNODE_FAB:
                                                                .BLKB   80
                                                00050 NML$A_NETNODE_RAB:
                                                                .BLKB   68
```

```
                        00094 NML$A_PROTECTION_XAB:
                                      .BLKB   88
                        000EC NML$A_SUMMARY_XAB:
                                      .BLKB   12
                        000F8 NML$A_NODE_ADDRESS_XAB:
                                      .BLKB   76
                        00144 NML$A_NODE_NAME_XAB:
                                      .BLKB   76
                        00190 NML$A_NODE_TYPE_XAB:
                                      .BLKB   76
                        001DC NML$A_NODE_LIST_XAB:
                                      .BLKB   76
                        00228 NML$T_KEY_VALUE:
                                      .BLKB   6

                                      .PSECT  $GLOBAL$,NOEXE,2

              00000007  00000 NML$GQ_NODE_FILE_DSC::
                                      .LONG   7
              00000000' 00004         .ADDRESS P.AAA

                              $RMS_PTR=          NML$A_SUMMARY_XAB
                                      .EXTRN   NML$_NODCVTERR, NML$GB_EVTSRCTYP
                                      .EXTRN   NML$GQ_EVTSRCDSC
                                      .EXTRN   NML$GW_EVTCLASS
                                      .EXTRN   NML$GB_EVTMSKTYP
                                      .EXTRN   NML$GQ_EVTMSKDSC
                                      .EXTRN   NML$GW_EVTSNKADR
                                      .EXTRN   NML$GW_ACP_CHAN
                                      .EXTRN   NML$GL_LOGMASK, NML$GQ_ENTSTRDSC
                                      .EXTRN   NML$AB_QIOBUFFER
                                      .EXTRN   NML$GQ_QIOBFDSC
                                      .EXTRN   NML$AB_EXEBUFFER
                                      .EXTRN   NML$GL_EXEDATPTR
                                      .EXTRN   NML$GQ_EXEDATDSC
                                      .EXTRN   NML$GQ_EXEBFDSC
                                      .EXTRN   NML$AB_RCVBUFFER
                                      .EXTRN   NML$GQ_RCVBFDSC
                                      .EXTRN   NML$AB_SNDBUFFER
                                      .EXTRN   NML$GQ_SNDBFDSC
                                      .EXTRN   NML$GL_RCVDATLEN
                                      .EXTRN   NML$AB_CPTABLE, NML$AB_MSGBLOCK
                                      .EXTRN   NML$AB_ENTITY_ID
                                      .EXTRN   NML$AB_QUALIFIER_ID
                                      .EXTRN   NML$AB_ENTITYDATA
                                      .EXTRN   NML$AB_NML_NMV, NML$AB_PRMSEM
                                      .EXTRN   NML$AB_RECBUF, NML$AL_ENTINFTAB
                                      .EXTRN   NML$AL_PERMINFTAB
                                      .EXTRN   NML$AW_PRM_DES, NML$GB_CMD_VER
                                      .EXTRN   NML$GB_ENTITY_CODE
                                      .EXTRN   NML$GB_ENTITY_FORMAT
                                      .EXTRN   NML$GL_QUALIFIER_PST
                                      .EXTRN   NML$GB_QUALIFIER_FORMAT
                                      .EXTRN   NML$GB_FUNCTION
                                      .EXTRN   NML$GB_INFO, NML$GB_OPTIONS
                                      .EXTRN   NML$GL_PRMCODE, NML$GL_PRS_FLGS
                                      .EXTRN   NML$GL_NML_ENTITY
```

```
                                                        .EXTRN  NML$GQ_NETNAMDSC
                                                        .EXTRN  NML$GQ_RECBFDSC
                                                        .EXTRN  NML$GW_PRMDESCNT
                                                        .EXTRN  NMA$SEARCHFLD, NML$CHKFILEIO
                                                        .EXTRN  NML$UPGRADE_PERM_DBS
                                                        .EXTRN  NML$DEBUG_TXT, NML$LOGFILEOP
                                                        .EXTRN  NML$LOGRECORDOP
                                                        .EXTRN  SYS$OPEN

                                                        .PSECT  $CODE$,NOWRT,2

                                        07FC 00000      .ENTRY  NML$OPEN_NODE_FILE, Save R2,R3,R4,R5,R6,R7,-; 0148
                                                                R8,R9,R10
                                5A 00000000G   00  9E 00002      MOVAB   SYS$OPEN, R10                          ;
                                59 00000000'   00  9E 00009      MOVAB   NML$GQ_NODE_FILE_DSC, R9               ;
                                58 00000000'   00  9E 00010      MOVAB   NML$A_SUMMARY_XAB, R8                  ;
                                57 00010001    8F  D0 00017      MOVL    #65537, STATUS                         ; 0169
                                     FF14      C8  9F 0001E      PUSHAB  NML$A_NETNODE_FAB                      ; 0170
                                56             6E  D0 00022      MOVL    FAB, R6                                ; 0171
                                         02    A6  B5 00025      TSTW    2(R6)                                  ;
                                               03  13 00028      BEQL    1$                                     ;
                                             00A6  31 0002A      BRW     5$                                     ;
    0050    8F          00          6E         00  2C 0002D 1$:  MOVC5   #0, (SP), #0, #80, (R6)               ; 0185
                                               66  00034                                                       ;
                                6E     5003    8F  B0 00035      MOVW    #20483, (R6)                           ;
                             16 A6     0F0F    8F  B0 0003A      MOVW    #3855, 22(R6)                          ;
                             1F A6             02  90 00040      MOVB    #2, 31(R6)                             ;
                             24 A6             68  9E 00044      MOVAB   NML$A_SUMMARY_XAB, 36(R6)              ;
                             2C A6         04  A9  D0 00048      MOVL    NML$GQ_NODE_FILE_DSC+4, 44(R6)         ;
                             30 A6 00000000'   00  9E 0004D      MOVAB   P.AAB, 48(R6)                          ;
                             34 A6             69  90 00055      MOVB    NML$GQ_NODE_FILE_DSC, 52(R6)           ;
                             35 A6             0F  90 00059      MOVB    #15, 53(R6)                            ;
        0C          00          6E             00  2C 0005D      MOVC5   #0, (SP), #0, #12, $RMS_PTR            ; 0186
                                               68  00062                                                       ;
                                68     0C16    8F  B0 00063      MOVW    #3094, $RMS_PTR                        ;
                                               56  DD 00068      PUSHL   R6                                     ; 0188
                                6A             01  FB 0006A      CALLS   #1, SYS$OPEN                           ;
                                57             50  D0 0006D      MOVL    R0, STATUS                             ;
                                2C             57  E9 00070      BLBC    STATUS, 2$                             ; 0189
                                01         09  A8  91 00073      CMPB    NML$A_SUMMARY_XAB+9, #1                ; 0195
                                               3D  12 00077      BNEQ    4$                                     ;
                 00000000V      00             00  FB 00079      CALLS   #0, NML$CLOSE_NODE_FILE                ; 0201
                 00000000G      00             00  FB 00080      CALLS   #0, NML$UPGRADE_PERM_DBS               ; 0208
                                57             50  D0 00087      MOVL    R0, STATUS                             ;
                                46             57  E9 0008A      BLBC    STATUS, 5$                             ; 0209
                     FF40       C8     04  A9  D0 0008D          MOVL    NML$GQ_NODE_FILE_DSC+4, -              ; 0211
                                                                    NML$A_NETNODE_FAB+44
                     FF48       C8             69  90 00093      MOVB    NML$GQ_NODE_FILE_DSC, NML$A_NETNODE_FAB+52 ; 0212
                                               56  DD 00098      PUSHL   R6                                     ; 0213
                                6A             01  FB 0009A      CALLS   #1, SYS$OPEN                           ;
                                               14  11 0009D      BRB     3$                                     ;
                 00018292       8F             57  D1 0009F 2$:  CMPL    STATUS, #98962                         ; 0222
                                               0E  12 000A6      BNEQ    4$                                     ;
                                         4200  8F  BB 000A8      PUSHR   #^M<R9,SP>                             ; 0223
                 00000000V      00             02  FB 000AC      CALLS   #2, NML$CREATE_NODE_DB                 ;
                                57             50  D0 000B3 3$:  MOVL    R0, STATUS                             ;
                                1A             57  E9 000B6 4$:  BLBC    STATUS, 5$                             ; 0229
```

```
                    00000000V  00              00  FB  000B9          CALLS   #0, NML$CONNECT_NODE_RAB                 ; 0231
                               57              50  D0  000C0          MOVL    R0, STATUS                              ;
                                     00000000'  00  9F  000C3          PUSHAB  P.AAC                                   ; 0234
                               7E              01  7D  000C9          MOVQ    #1, -(SP)                               ; 0232
                    00000000G  00              03  FB  000CC          CALLS   #3, NML$LOGFILEOP                       ;
                               50              57  D0  000D3  5$:      MOVL    STATUS, R0                              ; 0237
                                               04  000D6              RET                                             ; 0239

; Routine Size:  215 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  243          0240  1 %SBTTL 'nml$close_node_file  Close node permanent database file'
;  244          0241  1 GLOBAL ROUTINE nml$close_node_file  =
;  245          0242  1
;  246          0243  1 !++
;  247          0244  1 ! FUNCTIONAL DESCRIPTION:
;  248          0245  1 !
;  249          0246  1 !       This routine closes the node permanant database file.
;  250          0247  1 !
;  251          0248  1 ! FORMAL PARAMETERS:
;  252          0249  1 !       None
;  253          0250  1 !
;  254          0251  1 ! ROUTINE VALUE:
;  255          0252  1 ! COMPLETION CODES:
;  256          0253  1 !
;  257          0254  1 !       Failure or RMS error
;  258          0255  1 !
;  259          0256  1 !--
;  260          0257  1
;  261          0258  2 BEGIN
;  262          0259  2
;  263          0260  2 LOCAL
;  264          0261  2     fab :       REF BBLOCK,
;  265          0262  2     status;
;  266          0263  2
;  267          0264  2 status = nma$_success;
;  268          0265  2 !
;  269          0266  2 ! If the file isn't open, don't try to close it.
;  270          0267  2 !
;  271          0268  2 fab = nml$a_netnode_fab;
;  272          0269  2 IF .fab [fab$w_ifi] NEQ 0 THEN
;  273          0270  3     BEGIN
;  274          0271  3     status = $CLOSE (FAB = nml$a_netnode_fab);
;  275          0272  3     !
;  276          0273  3     ! If NML$LOG is defined with file io bit set, log a "file closed"
;  277          0274  3     ! message.
;  278          0275  3     !
;  279          0276  3     IF .status THEN
;  280          0277  3         nml$logfileop (dbg$c_fileio,
;  281          0278  3                        nma$c_opn_node,
;  282          0279  3                        $ASCID ('file closed'));
;  283          0280  3     END;
;  284          0281  2 RETURN .status;
;  285          0282  1 END;                ! of    nml$close_node_file


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                64  65  73  6F  6C  63  20  65  6C  69  66  0002C P.AAF:  .ASCII  \file closed\                    ;
                                                        00037           .BLKB   1
                                                0000000B 00038 P.AAE:   .LONG   11                              ;
                                                00000000' 0003C         .ADDRESS P.AAF                          ;

                                                                .EXTRN  SYS$CLOSE

                                                                .PSECT  $CODE$,NOWRT,2
```

```
                                        000C 00000          .ENTRY  NML$CLOSE_NODE_FILE, Save R2,R3                 ; 0241
                                53 00000000'    00   9E 00002       MOVAB   NML$A_NETNODE_FAB, R3                   ;
                                52              01   D0 00009       MOVL    #1, STATUS                              ; 0264
                                50              63   9E 0000C       MOVAB   NML$A_NETNODE_FAB, FAB                  ; 0268
                                        02      A0   B5 0000F       TSTW    2(FAB)                                  ; 0269
                                                1F   13 00012       BEQL    1$                                      ;
                                                53   DD 00014       PUSHL   R3                                      ; 0271
                        00000000G   00          01   FB 00016       CALLS   #1, SYS$CLOSE                           ;
                                52              50   D0 0001D       MOVL    R0, STATUS                              ;
                                10              52   E9 00020       BLBC    STATUS, 1$                              ; 0276
                                    00000000'   00   9F 00023       PUSHAB  P.AAE                                   ; 0279
                                7E              01   7D 00029       MOVQ    #1, -(SP)                               ; 0277
                        00000000G   00          03   FB 0002C       CALLS   #3, NML$LOGFILEOP                       ;
                                50              52   D0 00033 1$:    MOVL    STATUS, R0                              ; 0281
                                                04   00036         RET                                             ; 0282

; Routine Size:  55 bytes,    Routine Base:  $CODE$ + 00D7
```

```
;   287          0283  1 %SBTTL 'nml$read_node_rec   Get a Record in the Node File'
;   288          0284  1 GLOBAL ROUTINE nml$read_node_rec (key, key_value_dsc,
;   289          0285  1                                   node_type,
;   290          0286  1                                   buffer_dsc, data_dsc) =
;   291          0287  1
;   292          0288  1 !++
;   293          0289  1 ! FUNCTIONAL DESCRIPTION:
;   294          0290  1 !
;   295          0291  1 !       This routine performs $GETs to the node permanent database.  The
;   296          0292  1 !       database is organized with one record per node, four keys per
;   297          0293  1 !       record.  The four keys are:
;   298          0294  1 !                       node type (executor, remote node, loop node)
;   299          0295  1 !                       node address
;   300          0296  1 !                       node name
;   301          0297  1 !                       list node (node address concatenated with node type -
;   302          0298  1 !                               used for LISTing nodes).
;   303          0299  1 !
;   304          0300  1 ! FORMAL PARAMETERS:
;   305          0301  1 !
;   306          0302  1 !       key             key to use to identify the node's record.
;   307          0303  1 !       key_value_dsc   Descriptor of key value to use to identify the
;   308          0304  1 !                               node's record.
;   309          0305  1 !       node_type       Address for returning node type key value
;   310          0306  1 !       buffer_dsc      Address of a descriptor of a buffer to use
;   311          0307  1 !       data_dsc        Address of a descriptor to return descriptor of data
;   312          0308  1 !                               read.
;   313          0309  1 !
;   314          0310  1 ! ROUTINE VALUE:
;   315          0311  1 ! COMPLETION CODES:
;   316          0312  1 !
;   317          0313  1 !       NMA or RMS error status
;   318          0314  1 !
;   319          0315  1 !--
;   320          0316  1
;   321          0317  2 BEGIN
;   322          0318  2
;   323          0319  2 MAP
;   324          0320  2     buffer_dsc: REF VECTOR,             ! Buffer to use for record
;   325          0321  2     data_dsc: REF VECTOR;               ! Return data descriptor
;   326          0322  2
;   327          0323  2 LOCAL
;   328          0324  2     ptr:        REF BBLOCK,
;   329          0325  2     fab:        REF BBLOCK,
;   330          0326  2     rab:        REF BBLOCK,
;   331          0327  2     buf_ptr:    REF BBLOCK,
;   332          0328  2     local_dsc:  VECTOR [2],
;   333          0329  2     status;
;   334          0330  2
;   335          0331  2 fab = nml$a_netnode_fab;
;   336          0332  2 IF .fab [fab$w_ifi] EQL 0 THEN          ! If the node file isn't open
;   337          0333  2     RETURN .fab [fab$l_sts];            !       return open failure status.
;   338          0334  2 !
;   339          0335  2 ! Map the input key parameter to the key of reference number for that
;   340          0336  2 ! parameter.  If the key being used for this operation is different from the
;   341          0337  2 ! one the RAB is set up for, switch keys.
;   342          0338  2 !
;   343          0339  2 status = nml_map_keys (nmn$c_get_rec, .key, .key_value_dsc);
```

```
;  344          0340  2 IF .status THEN
;  345          0341  2     BEGIN
;  346          0342  2     rab = nml$a_netnode_rab;
;  347          0343  2     buf_ptr = .buffer_dsc [1];
;  348          0344  2     rab [rab$w_usz] = .buffer_dsc [0];
;  349          0345  2     rab [rab$l_ubf] = .buf_ptr;
;  350          0346  2
;  351          0347  2     status = $GET (RAB = .rab);
;  352          0348  2     END;
;  353          0349  2
;  354          0350  2 IF .status THEN
;  355          0351  2     BEGIN
;  356          0352  2     !
;  357          0353  2     ! Don't include keys in descriptor returned to caller.  Just return the
;  358          0354  2     ! NICE parameters and values.
;  359          0355  2     !
;  360          0356  2     data_dsc [0] = .rab [rab$w_rsz] - nmn$k_node_keys_len;
;  361          0357  2     data_dsc [1] = .buf_ptr + nmn$k_node_keys_len;
;  362          0358  2     !
;  363          0359  2     ! Return the node entity type since this is the only key that isn't
;  364          0360  2     ! duplicated in the NICE parameters.
;  365          0361  2     !
;  366          0362  3     .node_type =
;  367          0363  4         (SELECTONEU .buf_ptr [nmn$w_key_typ] OF
;  368          0364  4             SET
;  369          0365  4             [nmn$c_typ_remote]:         nml$c_node;
;  370          0366  4             [nmn$c_typ_exec]:           nml$c_executor;
;  371          0367  4             [nmn$c_typ_loopnode]:       nml$c_loopnode;
;  372          0368  4             TES);
;  373          0369  3     local_dsc [0] = .rab [rab$w_rsz];
;  374          0370  3     local_dsc [1] = .buf_ptr;
;  375          0371  3     nml$logrecordop (dbg$c_fileio,
;  376          0372  3                      nma$c_opn_node,
;  377          0373  3                      $ASCID ('record read'),
;  378          0374  3                      local_dsc);
;  379          0375  2     END;
;  380          0376  2 RETURN .status;
;  381          0377  1 END;                ! Of    nml$read_node_rec


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                64  61  65  72  20  64  72  6F  63  65  72  00040 P.AAH:  .ASCII  \record read\                          ;
                                                                00048       .BLKB   1
                                                    0000000B    0004C P:AAG:    .LONG   11                              ;
                                                    00000000'   00050           .ADDRESS P.AAH                          ;

                                                                .EXTRN  SYS$GET

                                                                .PSECT  $CODE$,NOWRT,2

                                                        001C    00000           .ENTRY  NML$READ_NODE_REC, Save R2,R3,R4 ; 0284
                                                    5E      08  C2  00002       SUBL2   #8, SP                          ;
                                                    50 00000000'    00  9E  00005   MOVAB   NML$A_NETNODE_FAB, FAB      ; 0331
                                                            02      A0  B5  0000C   TSTW    2(FAB)                      ; 0332
                                                                05  12  0000F       BNEQ    1$                          ;
```

```
                                50        08    A0  D0 00011          MOVL    8(FAB), R0                                  ; 0333
                                                    04 00015          RET
                                7E        04    AC  7D 00016 1$:     MOVQ    KEY, -(SP)                                  ; 0339
                                                    04  DD 0001A      PUSHL   #4
                     00000000V  00                  03  FB 0001C      CALLS   #3, NML_MAP_KEYS
                                54                  50  D0 00023      MOVL    R0, STATUS
                                78                  54  E9 00026      BLBC    STATUS, 6$                                  ; 0340
                                52  00000000'       00  9E 00029      MOVAB   NML$A_NETNODE_RAB, RAB                      ; 0342
                                50          10      AC  D0 00030      MOVL    BUFFER_DSC, R0                              ; 0343
                                53          04      A0  D0 00034      MOVL    4(R0), BUF_PTR
                           20   A2                  60  B0 00038      MOVW    (R0), 32(RAB)                               ; 0344
                           24   A2                  53  D0 0003C      MOVL    BUF_PTR, 36(RAB)                            ; 0345
                                                    52  DD 00040      PUSHL   RAB                                         ; 0347
                     00000000G  00                  01  FB 00042      CALLS   #1, SYS$GET
                                54                  50  D0 00049      MOVL    R0, STATUS
                                52                  54  E9 0004C      BLBC    STATUS, 6$                                  ; 0350
                                50          14      AC  D0 0004F      MOVL    DATA_DSC, R0                                ; 0356
                                60          22      A2  3C 00053      MOVZWL  34(RAB), (R0)
                                60                  0A  C2 00057      SUBL2   #10, (R0)
                           04   A0          0A      A3  9E 0005A      MOVAB   10(R3), 4(R0)                               ; 0357
                                50          02      A3  3C 0005F      MOVZWL  2(BUF_PTR), R0                              ; 0363
                                01                  50  B1 00063      CMPW    R0, #1                                      ; 0365
                                                    05  12 00066      BNEQ    2$
                                50                  03  D0 00068      MOVL    #3, R0
                                                    16  11 0006B      BRB     5$
                                                    50  D5 0006D 2$:   TSTL    R0                                          ; 0366
                                                    05  12 0006F      BNEQ    3$
                                50                  07  D0 00071      MOVL    #7, R0
                                                    0D  11 00074      BRB     5$
                                02                  50  B1 00076 3$:   CMPW    R0, #2                                      ; 0367
                                                    05  13 00079      BEQL    4$
                                50                  01  CE 0007B      MNEGL   #1, R0
                                                    03  11 0007E      BRB     5$
                                50                  05  D0 00080 4$:   MOVL    #5, R0
                           0C   BC                  50  D0 00083 5$:   MOVL    R0, @NODE_TYPE                              ; 0363
                                6E          22      A2  3C 00087      MOVZWL  34(RAB), LOCAL_DSC                          ; 0369
                           04   AE                  53  D0 0008B      MOVL    BUF_PTR, LOCAL_DSC+4                        ; 0370
                                                    5E  DD 0008F      PUSHL   SP                                          ; 0371
                                      00000000'     00  9F 00091      PUSHAB  P.AAG                                       ; 0373
                                7E                  01  7D 00097      MOVQ    #1, -(SP)                                   ; 0371
                     00000000G  00                  04  FB 0009A      CALLS   #4, NML$LOGRECORDOP
                                50                  54  D0 000A1 6$:   MOVL    STATUS, R0                                  ; 0376
                                                    04  000A4         RET                                                 ; 0377

; Routine Size:  165 bytes,    Routine Base:  $CODE$ + 010E
```

```
;   383          0378  1 %SBTTL 'nml$write_node_rec     Write a Record to the Node File'
;   384          0379  1 GLOBAL ROUTINE nml$write_node_rec (write_type, node_type, buffer_dsc) =
;   385          0380  1
;   386          0381  1 !++
;   387          0382  1 ! FUNCTIONAL DESCRIPTION:
;   388          0383  1 !
;   389          0384  1 !       This routine performs $PUTs to the node permanent database.  The
;   390          0385  1 !       database is organized with one record per node, four keys per
;   391          0386  1 !       record.  The four keys are:
;   392          0387  1 !                       node type (executor, remote node, loop node)
;   393          0388  1 !                       node address
;   394          0389  1 !                       node name
;   395          0390  1 !                       list node (node address concatenated with node type -
;   396          0391  1 !                               used for LISTing nodes in order by address).
;   397          0392  1 !
;   398          0393  1 ! FORMAL PARAMETERS:
;   399          0394  1 !       write_type      nmn$c_put_rec - do a $PUT
;   400          0395  1 !                       nmn$c_update_rec - do a $UPDATE
;   401          0396  1 !       node_type       Node entity type - in case it's changed.
;   402          0397  1 !       buffer_dsc      Address of a descriptor of the buffer to write.
;   403          0398  1 !                       This descriptor does not include the keys - only
;   404          0399  1 !                       the NICE parameters.
;   405          0400  1 !
;   406          0401  1 ! ROUTINE VALUE:
;   407          0402  1 ! COMPLETION CODES:
;   408          0403  1 !
;   409          0404  1 !       NMA or RMS error status
;   410          0405  1 !
;   411          0406  1 !--
;   412          0407  1
;   413          0408  2 BEGIN
;   414          0409  2
;   415          0410  2 MAP
;   416          0411  2     buffer_dsc: REF VECTOR;             ! Buffer to use for record
;   417          0412  2
;   418          0413  2 LOCAL
;   419          0414  2     buf_ptr:    REF BBLOCK,
;   420          0415  2     fab:        REF BBLOCK,
;   421          0416  2     rab:        REF BBLOCK,
;   422          0417  2     local_dsc:  VECTOR [2],
;   423          0418  2     param_dsc:  VECTOR [2],
;   424          0419  2     old_node_del_key,
;   425          0420  2     old_node_dsc:VECTOR [2],
;   426          0421  2     status;
;   427          0422  2
;   428          0423  2 fab = nml$a_netnode_fab;
;   429          0424  2 IF .fab [fab$w_ifi] EQL 0 THEN          ! If the node file isn't open
;   430          0425  2     RETURN .fab [fab$l_sts];           !       return open failure status.
;   431          0426  2 local_dsc [0] = .buffer_dsc [0] + nmn$k_node_keys_len;
;   432          0427  2 local_dsc [1] = .buffer_dsc [1] - nmn$k_node_keys_len;
;   433          0428  2 buf_ptr = .local_dsc [1];
;   434          0429  2 !
;   435          0430  2 ! First, get the node address from the NICE parameters in the permanent database
;   436          0431  2 ! record.  The node address is the primary key into the node permanent
;   437          0432  2 ! database.  Therefore, if it has changed the old record must be deleted
;   438          0433  2 ! before the new one can be written (since primary keys cannot be modified).
;   439          0434  2 !
```

```
;   440         0435  2 param_dsc [1] = 0;
;   441         0436  2 IF NOT nma$searchfld (.buffer_dsc, nma$c_pcno_add, param_dsc [0], param_dsc [1]) THEN
;   442         0437  2     param_dsc [1] = UPLIT (0);
;   443         0438  2 IF .buf_ptr [nmn$w_key_add] NEQ .(.param_dsc [1])<0,16> THEN
;   444         0439  3     BEGIN
;   445         0440  3     !
;   446         0441  3     ! If it's a brand new node, don't try to delete the old address's record.
;   447         0442  3     !
;   448         0443  3     IF .write_type NEQ nmn$c_put_rec THEN
;   449         0444  3         !
;   450         0445  3         ! It isn't a brand new node.  Delete the node using the address key if
;   451         0446  3         ! it's a remote node.  Use the type key if it's the exec - in case it
;   452         0447  3         ! has an address of 0 which could be confused with a loopnode.  Loopnodes
;   453         0448  3         ! never change addresses, so you never get here for loopnode operations.
;   454         0449  3         !
;   455         0450  4         BEGIN
;   456         0451  4         IF .buf_ptr [nmn$w_key_typ] EQL nmn$c_typ_exec THEN
;   457         0452  5             BEGIN
;   458         0453  5             old_node_del_key = nmn$c_typ_key_ref;
;   459         0454  5             old_node_dsc [0] = nmn$c_typ_key_len;
;   460         0455  5             old_node_dsc [1] = uplit (nml$c_executor);
;   461         0456  5             END
;   462         0457  4         ELSE
;   463         0458  5             BEGIN
;   464         0459  5             old_node_del_key = nma$c_pcno_add;
;   465         0460  5             old_node_dsc [0] = nmn$c_add_key_len;
;   466         0461  5             old_node_dsc [1] = .buf_ptr;
;   467         0462  4             END;
;   468         0463  4         nml$delete_node_rec (.old_node_del_key,
;   469         0464  4                              old_node_dsc);
;   470         0465  4         write_type = nmn$c_put_rec;
;   471         0466  3         END;
;   472         0467  3     buf_ptr [nmn$w_key_add] = .(.param_dsc [1]);        ! Put new address key
;   473         0468  3                                                         !       into record.
;   474         0469  2     END;
;   475         0470  2 !
;   476         0471  2 ! In case the node name, address or type has changed as a result of the
;   477         0472  2 ! NICE command being processed, change the corresponding key values as well.
;   478         0473  2 ! Now, get the node name from the NICE parameters.  If there isn't one,
;   479         0474  2 ! set up a null name.
;   480         0475  2 !
;   481         0476  2 param_dsc [1] = 0;
;   482         0477  2 IF nma$searchfld (.buffer_dsc, nma$c_pcno_nna, param_dsc [0], param_dsc [1]) THEN
;   483         0478  2     CH$COPY (.param_dsc [0], .param_dsc [1],
;   484         0479  2              %C' ',
;   485         0480  2              nmn$s_key_nam,
;   486         0481  2              buf_ptr [nmn$t_key_nam])
;   487         0482  2 ELSE
;   488         0483  2     CH$FILL (%C' ', nmn$s_key_nam, buf_ptr [nmn$t_key_nam]);
;   489         0484  2 !
;   490         0485  2 ! The third key is the node type.  The three node types are executor,
;   491         0486  2 ! remote, and loop node.
;   492         0487  2 !
;   493         0488  2 buf_ptr [nmn$w_key_typ] =
;   494         0489  3     (SELECTONE .node_type OF
;   495         0490  3         SET
;   496         0491  3         [nml$c_nodebyname, nml$c_node]: nmn$c_typ_remote;
```

```
3         [nml$c_executor]:              nmn$c_typ_exec;
3         [nml$c_loopnode]:              nmn$c_typ_loopnode;
2         TES);
2
2 !
2 ! Set up the buffer size and address to include the keys.
2 !
2 rab = nml$a_netnode_rab;
2 rab [rab$w_rsz] = .local_dsc [0];
2 rab [rab$l_rbf] = .local_dsc [1];
2
2 IF .write_type EQL nmn$c_put_rec THEN
3     status = $PUT (RAB = .rab)
2 ELSE
2     status = $UPDATE (RAB = .rab);
2
2 IF .status THEN
3     BEGIN
3     nml$logrecordop (dbg$c_fileio,
3                      nma$c_opn_node,
3                      $ASCID ('record written'),
3                      local_dsc);
2     END;
2 RETURN .status;
1 END;                    ! Of     nml$write_node_rec


                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2

                                   00000000  00054 P.AAI:  .LONG   0                                    ;
                                   00000007  00058 P.AAJ:  .LONG   7                                    ;
 6E 65 74 74 69 72 77 20 64 72 6F 63 65 72   0005C P.AAL:  .ASCII  \record written\                     ;
                                             0006A         .BLKB   2
                                   0000000E  0006C P.AAK:  .LONG   14                                   ;
                                   00000000' 00070         .ADDRESS P.AAL                               ;

                                                            .EXTRN  SYS$PUT, SYS$UPDATE

                                                            .PSECT  $CODE$,NOWRT,2

                                       01FC  00000         .ENTRY  NML$WRITE_NODE_REC, Save R2,R3,R4,R5,R6,R7,-; 0379
                                                                    R8                                   ;
                  58 00000000G   00   9E  00002         MOVAB   NMA$SEARCHFLD, R8                    ;
                  57 00000000'   00   9E  00009         MOVAB   P.AAI, R7                            ;
                  5E             18   C2  00010         SUBL2   #24, SP                              ;
                  50 00000000'   00   9E  00013         MOVAB   NML$A_NETNODE_FAB, FAB               ; 0423
                           02    A0   B5  0001A         TSTW    2(FAB)                               ; 0424
                                 05   12  0001D         BNEQ    1$                                   ;
                  50       08    A0   D0  0001F         MOVL    8(FAB), R0                           ; 0425
                                      04  00023         RET                                          ;
                  52       0C    AC   D0  00024 1$:     MOVL    BUFFER_DSC, R2                       ; 0426
      10  AE      62             0A   C1  00028         ADDL3   #10, (R2), LOCAL_DSC                 ;
      14  AE  04  A2             0A   C3  0002D         SUBL3   #10, 4(R2), LOCAL_DSC+4              ; 0427
                  56       14    AE   D0  00033         MOVL    LOCAL_DSC+4, BUF_PTR                 ; 0428
                           0C    AE   D4  00037         CLRL    PARAM_DSC+4                          ; 0435
                           0C    AE   9F  0003A         PUSHAB  PARAM_DSC+4                          ; 0436
```

```
                                   0C  AE  9F 0003D         PUSHAB  PARAM_DSC                              :
                         7E      01F6  8F  3C 00040         MOVZWL  #502, -(SP)                            :
                                       52  DD 00045         PUSHL   R2                                     :
                         68            04  FB 00047         CALLS   #4, NMA$SEARCHFLD                      :
                         04            50  E8 0004A         BLBS    R0, 2$                                 :
                    0C   AE            67  9E 0004D         MOVAB   P.AAI, PARAM_DSC+4                     : 0437
                    0C   BE            66  B1 00051 2$:     CMPW    (BUF_PTR), @PARAM_DSC+4                : 0438
                                       34  13 00055         BEQL    6$                                     :
                         01        04  AC  D1 00057         CMPL    WRITE_TYPE, #1                         : 0443
                                       2A  13 0005B         BEQL    5$                                     :
                         6E            02  D0 0005D         MOVL    #2, OLD_NODE_DSC                       : 0454
                                   02  A6  B5 00060         TSTW    2(BUF_PTR)                             : 0451
                                       0A  12 00063         BNEQ    3$                                     :
                         50            01  D0 00065         MOVL    #1, OLD_NODE_DEL_KEY                   : 0453
                    04   AE        04  A7  9E 00068         MOVAB   P.AAJ, OLD_NODE_DSC+4                  : 0455
                                       09  11 0006D         BRB     4$                                     : 0451
                         50      01F6  8F  3C 0006F 3$:     MOVZWL  #502, OLD_NODE_DEL_KEY                 : 0459
                    04   AE            56  D0 00074         MOVL    BUF_PTR, OLD_NODE_DSC+4                : 0461
                                 4001  8F  BB 00078 4$:     PUSHR   #^M<R0,SP>                             : 0463
             00000000V   00            02  FB 0007C         CALLS   #2, NML$DELETE_NODE_REC                :
                    04   AC            01  D0 00083         MOVL    #1, WRITE_TYPE                         : 0465
                         66        0C  BE  B0 00087 5$:     MOVW    @PARAM_DSC+4, (BUF_PTR)                : 0467
                                   0C  AE  D4 0008B 6$:     CLRL    PARAM_DSC+4                            : 0476
                                   0C  AE  9F 0008E         PUSHAB  PARAM_DSC+4                            : 0477
                                   0C  AE  9F 00091         PUSHAB  PARAM_DSC                              :
                         7E      01F4  8F  3C 00094         MOVZWL  #500, -(SP)                            :
                                       52  DD 00099         PUSHL   R2                                     :
                         68            04  FB 0009B         CALLS   #4, NMA$SEARCHFLD                      :
                         08            50  E9 0009E         BLBC    R0, 7$                                 :
     06          20     0C   BE        08  AE  2C 000A1     MOVC5   PARAM_DSC, @PARAM_DSC+4, #32, #6, -    : 0481
                                       04  A6     000A8             4(BUF_PTR)                             :
                                       07  11 000AA         BRB     8$                                     :
     06          20          6E            00  2C 000AC 7$: MOVC5   #0, (SP), #32, #6, 4(BUF_PTR)          : 0483
                                   04  A6     000B1                                                    :
                         50        08  AC  D0 000B3 8$:     MOVL    NODE_TYPE, R0                          : 0489
                         03            50  D1 000B7         CMPL    R0, #3                                 : 0491
                                       0A  1F 000BA         BLSSU   9$                                     :
                         04            50  D1 000BC         CMPL    R0, #4                                 :
                                       05  1A 000BF         BGTRU   9$                                     :
                         50            01  D0 000C1         MOVL    #1, R0                                 :
                                       16  11 000C4         BRB     12$                                    :
                         07            50  D1 000C6 9$:     CMPL    R0, #7                                 : 0492
                                       04  12 000C9         BNEQ    10$                                    :
                                       50  D4 000CB         CLRL    R0                                     :
                                       0D  11 000CD         BRB     12$                                    :
                         05            50  D1 000CF 10$:    CMPL    R0, #5                                 : 0493
                                       05  13 000D2         BEQL    11$                                    :
                         50            01  CE 000D4         MNEGL   #1, R0                                 :
                                       03  11 000D7         BRB     12$                                    :
                         50            02  D0 000D9 11$:    MOVL    #2, R0                                 :
                    02   A6            50  B0 000DC 12$:    MOVW    R0, 2(BUF_PTR)                         : 0489
                         50 00000000'  00  9E 000E0         MOVAB   NML$A_NETNODE_RAB, RAB                 : 0499
                    22   A0        10  AE  B0 000E7         MOVW    LOCAL_DSC, 34(RAB)                     : 0500
                    28   A0        14  AE  D0 000EC         MOVL    LOCAL_DSC+4, 40(RAB)                   : 0501
                         01        04  AC  D1 000F1         CMPL    WRITE_TYPE, #1                         : 0503
                                       08  12 000F5         BNEQ    13$                                    :
                                       50  DD 000F7         PUSHL   RAB                                    : 0504
```

```
                 00000000G  00          01  FB  000F9          CALLS   #1, SYS$PUT                             ;
                                        09  11  00100          BRB     14$                                     ;
                                        50  DD  00102  13$:     PUSHL   R0                                      ; 0506
                 00000000G  00          01  FB  00104          CALLS   #1, SYS$UPDATE                          ;
                            52          50  D0  0010B  14$:     MOVL    R0, STATUS                              ;
                            10          52  E9  0010E          BLBC    STATUS, 15$                             ; 0508
                                    10  AE  9F  00111          PUSHAB  LOCAL_DSC                               ; 0510
                                    18  A7  9F  00114          PUSHAB  P.AAK                                   ; 0512
                            7E          01  7D  00117          MOVQ    #1, -(SP)                               ; 0510
                 00000000G  00          04  FB  0011A          CALLS   #4, NML$LOGRECORDOP                     ;
                            50          52  D0  00121  15$:     MOVL    STATUS, R0                              ; 0515
                                        04  00124              RET                                             ; 0516

; Routine Size:  293 bytes,    Routine Base:  $CODE$ + 01B3
```

```
;   523     0517  1 %SBTTL 'nml$delete_node_rec     Delete a Record from the Node File'
;   524     0518  1 GLOBAL ROUTINE nml$delete_node_rec (key, key_value_dsc) =
;   525     0519  1
;   526     0520  1 !++
;   527     0521  1 ! FUNCTIONAL DESCRIPTION:
;   528     0522  1 !
;   529     0523  1 !       This routine performs $DELETEs on the node permanent database.  The
;   530     0524  1 !       database is organized with one record per node, four keys per
;   531     0525  1 !       record.  The four keys are:
;   532     0526  1 !                       node type (executor, remote node, loop node)
;   533     0527  1 !                       node address
;   534     0528  1 !                       node name
;   535     0529  1 !                       list node - node type concatenated with node address -
;   536     0530  1 !                               used for LISTing nodes.
;   537     0531  1 !
;   538     0532  1 ! FORMAL PARAMETERS:
;   539     0533  1 !
;   540     0534  1 !       key             Value mapped to the key of reference to use to
;   541     0535  1 !                       identify the node's record.
;   542     0536  1 !       key_value_dsc   Descriptor of key value to use to identify the
;   543     0537  1 !                               node's record.
;   544     0538  1 !
;   545     0539  1 ! ROUTINE VALUE:
;   546     0540  1 ! COMPLETION CODES:
;   547     0541  1 !
;   548     0542  1 !       NMA or RMS error status
;   549     0543  1 !
;   550     0544  1 !--
;   551     0545  1
;   552     0546  2 BEGIN
;   553     0547  2
;   554     0548  2 LOCAL
;   555     0549  2     rab: REF BBLOCK,
;   556     0550  2     status;
;   557     0551  2
;   558     0552  2 !
;   559     0553  2 ! Map the input key parameter to the key of reference number for that
;   560     0554  2 ! parameter.  If the key being used for this operation is different from the
;   561     0555  2 ! one the RAB is set up for, switch keys.
;   562     0556  2 !
;   563     0557  2 rab = nml$a_netnode_rab;
;   564     0558  2 status = rms$_suc;
;   565     0559  2 IF .key_value_dsc NEQ 0 THEN
;   566     0560  2     status = nml_map_keys (nmn$c_delete_rec, .key, .key_value_dsc);
;   567     0561  2 IF .status THEN
;   568     0562  2     status = $DELETE (RAB = .rab);
;   569     0563  2
;   570     0564  2 IF .status THEN
;   571     0565  3     BEGIN
;   572     0566  3     IF .key_value_dsc NEQ 0 THEN
;   573     0567  3         nml$logrecordop (dbg$c_fileio,
;   574     0568  3                          nma$c_opn_node,
;   575     0569  3                          $ASCID ('record deleted'),
;   576     0570  3                          .key_value_dsc)
;   577     0571  3     ELSE
;   578     0572  3         nml$debug_txt (dbg$c_fileio, $ASCID ('record deleted'));
;   579     0573  2     END;
```

```
;  580          0574  2 RETURN .status;
;  581          0575  1 END;                 ! Of    nml$delete_node_rec



                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

    64  65  74  65  6C  65  64  20  64  72  6F  63  65  72  00074 P.AAN:  .ASCII  \record deleted\                                ;
                                                                00082       .BLKB   2
                                                          0000000E  00084 P.AAM:  .LONG   14                                    ;
                                                          00000000' 00088       .ADDRESS P.AAN                                  ;
    64  65  74  65  6C  65  64  20  64  72  6F  63  65  72 0008C P.AAP:  .ASCII  \record deleted\                                ;
                                                                0009A       .BLKB   2
                                                          0000000E  0009C P.AAO:  .LONG   14                                    ;
                                                          00000000' 000A0       .ADDRESS P.AAP                                  ;

                                                                        .EXTRN  SYS$DELETE

                                                                        .PSECT  $CODE$,NOWRT,2

                                                              003C 00000        .ENTRY  NML$DELETE_NODE_REC, Save R2,R3,R4,R5       ; 0518
                                                55 00000000'   00   9E 00002    MOVAB   NML$A_NETNODE_RAB, RAB                      ; 0557
                                                54 00010001    8F   D0 00009    MOVL    #65537, STATUS                              ; 0558
                                                52         08  AC   D0 00010    MOVL    KEY_VALUE_DSC, R2                           ; 0559
                                                               53   D4 00014    CLRL    R3                                          ;
                                                               52   D5 00016    TSTL    R2                                          ;
                                                               13   13 00018    BEQL    1$                                          ;
                                                               53   D6 0001A    INCL    R3                                          ;
                                                               52   DD 0001C    PUSHL   R2                                          ; 0560
                                                           04  AC   DD 0001E    PUSHL   KEY                                         ;
                                                               03   DD 00021    PUSHL   #3                                          ;
                                                00000000V  00   03   FB 00023    CALLS   #3, NML_MAP_KEYS                            ;
                                                           54       50   D0 0002A    MOVL    R0, STATUS                              ;
                                                           35       54   E9 0002D 1$: BLBC    STATUS, 3$                              ; 0561
                                                               55   DD 00030    PUSHL   RAB                                         ; 0562
                                                00000000G  00   01   FB 00032    CALLS   #1, SYS$DELETE                              ;
                                                           54       50   D0 00039    MOVL    R0, STATUS                              ;
                                                           26       54   E9 0003C    BLBC    STATUS, 3$                              ; 0564
                                                           14       53   E9 0003F    BLBC    R3, 2$                                  ; 0566
                                                               52   DD 00042    PUSHL   R2                                          ; 0570
                                                    00000000'  00   9F 00044    PUSHAB  P.AAM                                       ; 0569
                                                           7E       01   7D 0004A    MOVQ    #1, -(SP)                               ; 0567
                                                00000000G  00   04   FB 0004D    CALLS   #4, NML$LOGRECORDOP                         ;
                                                               0F   11 00054    BRB     3$                                          ;
                                                    00000000'  00   9F 00056 2$: PUSHAB  P.AAO                                       ; 0572
                                                               01   DD 0005C    PUSHL   #1                                          ;
                                                00000000G  00   02   FB 0005E    CALLS   #2, NML$DEBUG_TXT                           ;
                                                           50       54   D0 00065 3$: MOVL    STATUS, R0                              ; 0574
                                                                    04   00068    RET                                             ; 0575

; Routine Size:  105 bytes,    Routine Base:  $CODE$ + 02D8
```

```
;  583          0576  1 %SBTTL 'nml_map_keys              Switch key used to access node database'
;  584          0577  1 ROUTINE nml_map_keys (function, key_param, key_value_dsc) =
;  585          0578  1
;  586          0579  1 !++
;  587          0580  1 ! FUNCTIONAL DESCRIPTION:
;  588          0581  1 !       This routine is called whenever a record in the node permanent
;  589          0582  1 !       database is accessed.  It sets up the key reference, length, and
;  590          0583  1 !       value so the next RMS operation is done on the correct record.
;  591          0584  1 !
;  592          0585  1 ! FORMAL PARAMETERS:
;  593          0586  1 !       function        nmn$c_put_rec = doing a put.
;  594          0587  1 !                       nmn$c_get_rec = doing a read.
;  595          0588  1 !                       nmn$c_delete_rec = deleteing a record.
;  596          0589  1 !                       nmn$c_update_rec = updating a record.
;  597          0590  1 !       key_param       Value mapped to the key of reference to use to
;  598          0591  1 !                       identify the node's record.
;  599          0592  1 !       key_value_dsc   Descriptor of key value to use to identify the
;  600          0593  1 !                               node's record.
;  601          0594  1 !
;  602          0595  1 ! ROUTINE VALUE:
;  603          0596  1 ! COMPLETION CODES:
;  604          0597  1 !
;  605          0598  1 !       Failure or RMS error
;  606          0599  1 !
;  607          0600  1 !--
;  608          0601  1
;  609          0602  2 BEGIN
;  610          0603  2
;  611          0604  2 MAP
;  612          0605  2     key_value_dsc: REF VECTOR;          ! Descriptor for key value
;  613          0606  2
;  614          0607  2 LOCAL
;  615          0608  2     rab: REF BBLOCK,
;  616          0609  2     fab: REF BBLOCK,
;  617          0610  2     key_ref,
;  618          0611  2     key_addr,
;  619          0612  2     key_len,
;  620          0613  2     name_buf: BBLOCK [nmn$c_nam_key_len],
;  621          0614  2     do_find,
;  622          0615  2     status;
;  623          0616  2
;  624          0617  2 rab = nml$a_netnode_rab;
;  625          0618  2 fab = nml$a_netnode_fab;
;  626          0619  2 IF .fab [fab$w_ifi] EQL 0 THEN           ! If the node file isn't open
;  627          0620  2     status = .fab [fab$l_sts]           !       return open failure status.
;  628          0621  2 ELSE
;  629          0622  3     BEGIN
;  630          0623  3     !
;  631          0624  3     ! Fill in key value.  This identifies the specific node record to get, put,
;  632          0625  3     ! or delete.  Also, set up the buffer size and address.
;  633          0626  3     !
;  634          0627  3     key_len = .key_value_dsc [0];
;  635          0628  3     rab [rab$l_kbf] = nml$t_key_value;
;  636          0629  3     rab [rab$v_kge] = 0;
;  637          0630  3     SELECTONEU .key_param OF
;  638          0631  3         SET
;  639          0632  3         !
```

```
;  640          0633  3          ! If the key is list node or node type, map it to the key values used
;  641          0634  3          ! in the node database file.  The value is passed to this routine as
;  642          0635  3          ! an 'NML$C_' node entity type.  The list key overlaps with the node
;  643          0636  3          ! address key to allow the LIST command to get nodes by type and,
;  644          0637  3          ! within type, sequentially by address.  The list key value contains
;  645          0638  3          ! a zero for the node address; hence when you do a $GET of (type OR 0)
;  646          0639  3          ! with a match type of GTR, it will get the first node of that type
;  647          0640  3          ! in the file.  Subsequent sequential reads will return the nodes of
;  648          0641  3          ! that type in ascending order by address.
;  649          0642  3          !
;  650          0643  3          [nmn$c_typ_key_ref,nmn$c_lis_key_ref]:
;  651          0644  4              BEGIN
;  652          0645  5              key_addr = (SELECTONEU .(.key_value_dsc [1]) OF
;  653          0646  5                  SET
;  654          0647  5                  [nml$c_nodebyname,
;  655          0648  5                   nml$c_node]:           UPLIT WORD (0, nmn$c_typ_remote);
;  656          0649  5                  [nml$c_executor]:      UPLIT WORD (0, nmn$c_typ_exec);
;  657          0650  5                  [nml$c_loopnode]:      UPLIT WORD (0, nmn$c_typ_loopnode);
;  658          0651  4                  TES);
;  659          0652  4              IF .key_param EQL nmn$c_typ_key_ref THEN
;  660          0653  4                  key_addr = .key_addr + 2
;  661          0654  4              ELSE
;  662          0655  4                  rab [rab$v_kge] = 1;
;  663          0656  4              key_ref = .key_param;
;  664          0657  3              END;
;  665          0658  3          [nma$c_pcno_add]:
;  666          0659  4              BEGIN
;  667          0660  4              key_ref = nmn$c_add_key_ref;
;  668          0661  4              key_addr = .key_value_dsc [1];
;  669          0662  3              END;
;  670          0663  3          [nma$c_pcno_nna]:
;  671          0664  4              BEGIN
;  672          0665  4              key_ref = nmn$c_nam_key_ref;
;  673          0666  4              key_addr = name_buf;
;  674          0667  4              key_len = nmn$c_nam_key_len;
;  675          0668  4              CH$COPY (.key_value_dsc [0], .key_value_dsc [1], %C' ',
;  676          0669  4                  nmn$c_nam_key_len, name_buf);
;  677          0670  3              END;
;  678          0671  3          TES;
;  679          0672  3          !
;  680          0673  3          ! If doing an update or delete operation, check to see if the
;  681          0674  3          ! key from the last operation is different (DEF EXEC NAME requires
;  682          0675  3          ! that the name be checked, so an intermediate read is done between
;  683          0676  3          ! the $GET of the executor node entry, and the $UPDATE).  If the key
;  684          0677  3          ! is different, do a $FIND so that RMS has the correct current record
;  685          0678  3          ! for the update or delete.
;  686          0679  3          !
;  687          0680  3          IF .function EQL nmn$c_update_rec OR
;  688          0681  3             .function EQL nmn$c_delete_rec THEN
;  689          0682  4              BEGIN
;  690          0683  4              IF .key_ref NEQ .rab [rab$b_krf] OR
;  691          0684  4                 CH$NEQ (.key_len, .key_addr,
;  692          0685  4                         .rab [rab$b_ksz], .rab [rab$l_kbf], %C' ') THEN
;  693          0686  4                  do_find = true
;  694          0687  4              ELSE
;  695          0688  4                  do_find = false;
;  696          0689  3              END;
```

```
;  697     0690  3     !
;  698     0691  3     ! Put the new key reference, key size, and key value into the RAB.  These
;  699     0692  3     ! are the fields that identify the node record to RMS.
;  700     0693  3     !
;  701     0694  3     rab [rab$b_krf] = .key_ref;
;  702     0695  3     rab [rab$b_ksz] = .key_len;
;  703     0696  3     rab [rab$l_kbf] = nml$t_key_value;
;  704     0697  3     CH$MOVE (.key_len, .key_addr, nml$t_key_value);
;  705     0698  3     status = rms$_suc;
;  706     0699  3     IF .do_find THEN
;  707     0700  3         status = $FIND (RAB = .rab);
;  708     0701  2     END;
;  709     0702  2 RETURN .status;
;  710     0703  1 END;                ! of   nml_map_keys


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                    0001  0000  000A4 P.AAQ:   .WORD   0, 1                                          ;
                                    0000  0000  000A8 P.AAR:   .WORD   0, 0                                          ;
                                    0002  0000  000AC P.AAS:   .WORD   0, 2                                          ;

                                                        .EXTRN  SYS$FIND

                                                        .PSECT  $CODE$,NOWRT,2

                                          0FFC 00000 NML_MAP_KEYS:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                ; 0577
                                 5E             08  C2 00002          SUBL2   #8, SP                               ;
                                 56 00000000'   00  9E 00005          MOVAB   NML$A_NETNODE_RAB, RAB               ; 0617
                                 50 00000000'   00  9E 0000C          MOVAB   NML$A_NETNODE_FAB, FAB               ; 0618
                                       02       A0  B5 00013          TSTW    2(FAB)                               ; 0619
                                                07  12 00016          BNEQ    1$                                   ;
                                 5B             08  A0 00018          MOVL    8(FAB), STATUS                       ; 0620
                                                00EA 31 0001C          BRW     16$                                  ;
                                 50             0C  AC D0 0001F 1$:    MOVL    KEY_VALUE_DSC, R0                    ; 0627
                                 5A                 60 D0 00023          MOVL    (R0), KEY_LEN                        ;
                              30 A6 00000000'       00 9E 00026          MOVAB   NML$T_KEY_VALUE, 48(RAB)             ; 0628
                              06 A6                 20 8A 0002E          BICB2   #32, 6(RAB)                          ; 0629
                                 52             08  AC D0 00032          MOVL    KEY_PARAM, R2                        ; 0630
                                 01                 52 D1 00036          CMPL    R2, #1                               ; 0643
                                                    05 13 00039          BEQL    2$                                   ;
                                 03                 52 D1 0003B          CMPL    R2, #3                               ;
                                                    49 12 0003E          BNEQ    9$                                   ;
                                 51             04  B0 D0 00040 2$:    MOVL    @4(R0), R1                           ; 0645
                                 03                 51 D1 00044          CMPL    R1, #3                               ; 0647
                                                    0E 1F 00047          BLSSU   3$                                   ;
                                 04                 51 D1 00049          CMPL    R1, #4                               ;
                                                    09 1A 0004C          BGTRU   3$                                   ;
                                 57 00000000'       00 9E 0004E          MOVAB   P.AAQ, KEY_ADDR                      ; 0648
                                                    1F 11 00055          BRB     6$                                   ;
                                 07                 51 D1 00057 3$:      CMPL    R1, #7                               ; 0649
                                                    09 12 0005A          BNEQ    4$                                   ;
                                 57 00000000'       00 9E 0005C          MOVAB   P.AAR, KEY_ADDR                      ;
                                                    11 11 00063          BRB     6$                                   ;
                                 05                 51 D1 00065 4$:      CMPL    R1, #5                               ; 0650
```

```
                                        05  13 00068          BEQL    5$                                              ;
                              57        01  CE 0006A          MNEGL   #1, KEY_ADDR                                    ;
                                        07  11 0006D          BRB     6$                                              ;
                              57 00000000'  00  9E 0006F 5$:  MOVAB   P.AAS, KEY_ADDR                                 ;
                              01        52  D1 00076 6$:      CMPL    R2, #1                                          ; 0652
                                        05  12 00079          BNEQ    7$                                              ;
                              57        02  C0 0007B          ADDL2   #2, KEY_ADDR                                    ; 0653
                                        04  11 0007E          BRB     8$                                              ;
                         06   A6        20  88 00080 7$:      BISB2   #32, 6(RAB)                                     ; 0655
                              58        52  D0 00084 8$:      MOVL    R2, KEY_REF                                     ; 0656
                                        2A  11 00087          BRB     11$                                             ; 0630
                   000001F6   8F        52  D1 00089 9$:      CMPL    R2, #502                                        ; 0658
                                        08  12 00090          BNEQ    10$                                             ;
                                        58  D4 00092          CLRL    KEY_REF                                         ; 0660
                              57     04 A0  D0 00094          MOVL    4(R0), KEY_ADDR                                 ; 0661
                                        19  11 00098          BRB     11$                                             ; 0630
                   000001F4   8F        52  D1 0009A 10$:     CMPL    R2, #500                                        ; 0663
                                        10  12 000A1          BNEQ    11$                                             ;
                              58        02  D0 000A3          MOVL    #2, KEY_REF                                     ; 0665
                              57        6E  9E 000A6          MOVAB   NAME_BUF, KEY_ADDR                              ; 0666
                              5A        06  D0 000A9          MOVL    #6, KEY_LEN                                     ; 0667
      06            20     04 B0        60  2C 000AC          MOVC5   (R0), @4(R0), #32, #6, NAME_BUF                 ; 0668
                                        6E     000B2                                                          ;
                              02     04 AC  D1 000B3 11$:     CMPL    FUNCTION, #2                                    ; 0680
                                        06  13 000B7          BEQL    12$                                             ;
                              03     04 AC  D1 000B9          CMPL    FUNCTION, #3                                    ; 0681
                                        1C  12 000BD          BNEQ    15$                                             ;
      58       35   A6        08        00  ED 000BF 12$:     CMPZV   #0, #8, 53(RAB), KEY_REF                        ; 0683
                                        0D  12 000C5          BNEQ    13$                                             ;
                              50     34 A6  9A 000C7          MOVZBL  52(RAB), R0                                     ; 0685
      50            20        67        5A  2D 000CB          CMPC5   KEY_LEN, (KEY_ADDR), #32, R0, @48(RAB)          ; 0684
                                     30 B6     000D0                                                          ;
                                        05  13 000D2          BEQL    14$                                             ;
                              59        01  D0 000D4 13$:     MOVL    #1, DO_FIND                                     ; 0686
                                        02  11 000D7          BRB     15$                                             ;
                                        59  D4 000D9 14$:     CLRL    DO_FIND                                         ; 0688
                         35   A6        58  90 000DB 15$:     MOVB    KEY_REF, 53(RAB)                                ; 0694
                         34   A6        5A  90 000DF          MOVB    KEY_LEN, 52(RAB)                                ; 0695
                         30   A6 00000000'  00  9E 000E3      MOVAB   NML$T_KEY_VALUE, 48(RAB)                        ; 0696
         00000000'  00        67        5A  28 000EB          MOVC3   KEY_LEN, (KEY_ADDR), NML$T_KEY_VALUE            ; 0697
                              5B 00010001   8F  D0 000F3      MOVL    #65537, STATUS                                  ; 0698
                              0C        59  E9 000FA          BLBC    DO_FIND, 16$                                    ; 0699
                                        56  DD 000FD          PUSHL   RAB                                             ; 0700
                   00000000G  00        01  FB 000FF          CALLS   #1, SYS$FIND                                    ;
                              5B        50  D0 00106          MOVL    R0, STATUS                                      ;
                              50        5B  D0 00109 16$:     MOVL    STATUS, R0                                      ; 0702
                                        04     0010C          RET                                                     ; 0703

; Routine Size:  269 bytes,    Routine Base:  $CODE$ + 0341
```

```
;   712          0704  1 %SBTTL 'nml$read_loopnode       Get a loopnode in the Node File'
;   713          0705  1 GLOBAL ROUTINE nml$read_loopnode (the_circuit_dsc,
;   714          0706  1                                  buffer_dsc, data_dsc) =
;   715          0707  1
;   716          0708  1 !++
;   717          0709  1 ! FUNCTIONAL DESCRIPTION:
;   718          0710  1 !
;   719          0711  1 !       This routine searches through the node permanent database for
;   720          0712  1 !       a loopnode on the specified circuit.  Loopnodes must be set up
;   721          0713  1 !       with unique circuit ids.
;   722          0714  1 !       This routine is called for such functions as:
;   723          0715  1 !               LIST CIRCUIT    - in case the circuit is set up as a loopnode,
;   724          0716  1 !                                 to get the loopnode name.
;   725          0717  1 !               DEFINE NODE node-id CIRCUIT circuit-id   - to make sure there
;   726          0718  1 !                                 isn't already a loopnode on that circuit.
;   727          0719  1 !
;   728          0720  1 ! FORMAL PARAMETERS:
;   729          0721  1 !
;   730          0722  1 !       the_circuit_dsc Address of descriptor of circuit ID to look for.
;   731          0723  1 !       buffer_dsc      Address of a descriptor of a buffer to use for
;   732          0724  1 !                       returning the loopnode data.
;   733          0725  1 !       data_dsc        Address of a descriptor to return descriptor of data
;   734          0726  1 !                       read.
;   735          0727  1 !
;   736          0728  1 ! ROUTINE VALUE:
;   737          0729  1 ! COMPLETION CODES:
;   738          0730  1 !
;   739          0731  1 !       NMA or RMS error status
;   740          0732  1 !
;   741          0733  1 !--
;   742          0734  1
;   743          0735  2 BEGIN
;   744          0736  2
;   745          0737  2 MAP
;   746          0738  2     the_circuit_dsc: REF VECTOR;
;   747          0739  2
;   748          0740  2 LOCAL
;   749          0741  2     a_circuit_dsc: VECTOR [2],
;   750          0742  2     rewind_flag,
;   751          0743  2     status;
;   752          0744  2
;   753          0745  2 !
;   754          0746  2 ! Read through the known loopnodes in the node permanent database, looking
;   755          0747  2 ! for a loopnode on the circuit specified by the input parameter.
;   756          0748  2 !
;   757          0749  2 rewind_flag = true;
;   758          0750  2 WHILE status = nml$read_known_node_rec (nml$c_loopnode,
;   759          0751  2                                        .buffer_dsc,
;   760          0752  2                                        .data_dsc
;   761          0753  2                                        .rewind_flag) DO
;   762          0754  2
;   763          0755  3     BEGIN
;   764          0756  3     rewind_flag = false;
;   765          0757  3     a_circuit_dsc [0] = 0;
;   766          0758  3     a_circuit_dsc [1] = 0;
;   767          0759  3     !
;   768          0760  3     ! Find the circuit ID for this loopnode, and, if it matches the
```

```
;  769          0761  3     ! circuit I'm looking for, return the loopnode data to the caller.
;  770          0762  3     !
;  771          0763  3     IF nma$searchfld (.data_dsc,
;  772          0764  3                       nma$c_pcno_nli,
;  773          0765  3                       a_circuit_dsc [0],
;  774          0766  3                       a_circuit_dsc [1]) AND
;  775          0767  3         CH$EQL (.the_circuit_dsc [0], .the_circuit_dsc [1],
;  776          0768  3                 .a_circuit_dsc [0], .a_circuit_dsc [1]) THEN
;  777          0769  3         EXITLOOP;
;  778          0770  2     END;
;  779          0771  2 RETURN .status;
;  780          0772  1 END;                ! of nml$read_loopnode


                                                003C 00000          .ENTRY  NML$READ_LOOPNODE, Save R2,R3,R4,R5                 ; 0705
                                           5E   08  C2 00002          SUBL2   #8, SP                                              ;
                                           54   01  D0 00005          MOVL    #1, REWIND_FLAG                                     ; 0749
                                                54  DD 00008 1$:       PUSHL   REWIND_FLAG                                         ; 0753
                                           7E       AC  7D 0000A       MOVQ    BUFFER_DSC, -(SP)                                   ; 0751
                                                05  DD 0000E          PUSHL   #5                                                  ; 0750
                                 00000000V 00       04  FB 00010       CALLS   #4, NML$READ_KNOWN_NODE_REC                         ;
                                           55       50  D0 00017       MOVL    R0, STATUS                                          ;
                                           2A       55  E9 0001A       BLBC    STATUS, 2$                                          ;
                                                54  D4 0001D          CLRL    REWIND_FLAG                                         ; 0756
                                                6E  7C 0001F          CLRQ    A_CIRCUIT_DSC                                       ; 0757
                                              04    AE  9F 00021       PUSHAB  A_CIRCUIT_DSC+4                                     ; 0766
                                              04    AE  9F 00024       PUSHAB  A_CIRCUIT_DSC                                       ; 0765
                                           7E   01F5  8F  3C 00027       MOVZWL  #501, -(SP)                                         ; 0763
                                                0C    AC  DD 0002C       PUSHL   DATA_DSC                                            ;
                                    00000000G   00      04  FB 0002F    CALLS   #4, NMA$SEARCHFLD                                   ;
                                           CF       50  E9 00036       BLBC    R0, 1$                                              ;
                                           50       04  AC  D0 00039       MOVL    THE_CIRCUIT_DSC, R0                             ; 0767
                                 6E        00       04  B0  60  2D 0003D    CMPC5   (R0), @4(R0), #0, A_CIRCUIT_DSC, -          ;
                                                04    BE      00043                  @A_CIRCUIT_DSC+4                                ;
                                                C1  12 00045          BNEQ    1$                                                  ;
                                           50       55  D0 00047 2$:       MOVL    STATUS, R0                                      ; 0771
                                                    04  0004A          RET                                                         ; 0772

; Routine Size:  75 bytes,    Routine Base:  $CODE$ + 044E
```

```
;  782          0773  1 %SBTTL 'nml$read_known_node_rec   Get a known Record in the Node File'
;  783          0774  1 GLOBAL ROUTINE nml$read_known_node_rec (node_type,
;  784          0775  1                                        buffer_dsc,
;  785          0776  1                                        data_dsc
;  786          0777  1                                        rewind_flag) =
;  787          0778  1
;  788          0779  1 !++
;  789          0780  1 ! FUNCTIONAL DESCRIPTION:
;  790          0781  1 !
;  791          0782  1 !       This routine performs sequential $GETs to the node permanent
;  792          0783  1 !       database.  The  database is organized with one record per node,
;  793          0784  1 !       The four keys are:
;  794          0785  1 !                       node type (executor, remote node, loop node)
;  795          0786  1 !                       node address
;  796          0787  1 !                       node name
;  797          0788  1 !                       list node - node type concatenated with node address -
;  798          0789  1 !                               used for LISTing nodes.
;  799          0790  1 !       If the node key and value are different from the last time
;  800          0791  1 !       this routine was called, do the $GET with a record access mode
;  801          0792  1 !       of keyed.  If they are the same, do the $GET with a record access
;  802          0793  1 !       mode of sequential.  The latter will cause RMS to return the
;  803          0794  1 !       next record in the file greater which matches the key and is
;  804          0795  1 !       greater than the key value.  This is useful for KNOWN NODES and
;  805          0796  1 !       KNOWN LOOPNODES operations.
;  806          0797  1 !
;  807          0798  1 ! FORMAL PARAMETERS:
;  808          0799  1 !
;  809          0800  1 !       node_type       Node entity type
;  810          0801  1 !       buffer_dsc      Address of a descriptor of a buffer to use
;  811          0802  1 !       data_dsc        Address of a descriptor to return descriptor of data
;  812          0803  1 !                               read.
;  813          0804  1 !       rewind_flag     Set if the caller wants to begin reading at the
;  814          0805  1 !                       beginning of the node file.
;  815          0806  1 !
;  816          0807  1 ! ROUTINE VALUE:
;  817          0808  1 ! COMPLETION CODES:
;  818          0809  1 !
;  819          0810  1 !       NMA or RMS error status
;  820          0811  1 !
;  821          0812  1 !--
;  822          0813  1
;  823          0814  2 BEGIN
;  824          0815  2
;  825          0816  2 MAP
;  826          0817  2     buffer_dsc: REF VECTOR,             ! Buffer to use for record
;  827          0818  2     data_dsc: REF VECTOR;               ! Return data descriptor
;  828          0819  2
;  829          0820  2 LOCAL
;  830          0821  2     rab: REF BBLOCK,
;  831          0822  2     key_value_dsc:      VECTOR [2],     ! Descriptor for key value
;  832          0823  2     rec_node_type,
;  833          0824  2     status;
;  834          0825  2
;  835          0826  2 OWN
;  836          0827  2     last_RFA0,                          ! Record file address of last record
;  837          0828  2     last_RFA4 : WORD;                   !       read by this routine.
;  838          0829  2
```

```
;  839          0830  2 rab = nml$a_netnode_rab;
;  840          0831  2 key_value_dsc [0] = nmn$c_lis_key_len;
;  841          0832  2 key_value_dsc [1] = node_type;
;  842          0833  2 status = nml$_sts_suc;
;  843          0834  2 !
;  844          0835  2 ! Known nodes are found using the Type and Address keys with a search type
;  845          0836  2 ! of "greater than or equal to".  If the last operation was to a node in the
;  846          0837  2 ! middle of the type being LISTed, RMS's "next record" will cause it to start
;  847          0838  2 ! reading node records from there.  So, do a $REWIND so RMS starts at the
;  848          0839  2 ! beginning of the file.
;  849          0840  2 !
;  850          0841  2 IF .rewind_flag THEN
;  851          0842  3     BEGIN
;  852          0843  3     last_RFA0 = 0;
;  853          0844  3     last_RFA4 = 0;
;  854          0845  3     status = $REWIND (RAB = .rab);
;  855          0846  3     END;
;  856          0847  2 IF .status THEN
;  857          0848  3     BEGIN
;  858          0849  3     !
;  859          0850  3     ! If this is the second (or later) time this routine is being called to
;  860          0851  3     ! find a node record, set up the RAB to do the next read sequentially.
;  861          0852  3     !
;  862          0853  3     IF NOT .rewind_flag THEN
;  863          0854  4         BEGIN
;  864          0855  4         !
;  865          0856  4         ! Some operations, such as LIST KNOWN NODES CHARACTERISTICS, must
;  866          0857  4         ! read random node records between the sequential operations done
;  867          0858  4         ! by this routine.  For example, when listing a node which has the HOST
;  868          0859  4         ! parameter set, the HOST node's record must be read in to determine
;  869          0860  4         ! the host node's name to include in the LIST response.  If the Record
;  870          0861  4         ! File Address in the RAB has moved, do a $GET to get back to where
;  871          0862  4         ! we were.
;  872          0863  4         !
;  873          0864  4         ! PURGE KNOWN NODES ALL deletes a record between each call to this
;  874          0865  4         ! routine.  In this case the RFA is zeroed, so check for that too
;  875          0866  4         ! before doing the $GET.
;  876          0867  4         !
;  877          0868  5         IF (.last_RFA0 NEQ .rab [rab$l_rfa0] OR
;  878          0869  5            .last_RFA4 NEQ .rab [rab$w_rfa4])
;  879          0870  4                       AND
;  880          0871  5            (.rab [rab$l_rfa0] NEQ 0 OR
;  881          0872  5             .rab [rab$w_rfa4] NEQ 0)
;  882          0873  4         THEN
;  883          0874  5             BEGIN
;  884          0875  5             rab [rab$b_rac] = rab$c_rfa;
;  885          0876  5             rab [rab$l_rfa0] = .last_RFA0;
;  886          0877  5             rab [rab$w_rfa4] = .last_RFA4;
;  887          0878  5             rab [rab$w_usz] = .buffer_dsc [0];
;  888          0879  5             rab [rab$l_ubf] = .buffer_dsc [1];
;  889          0880  5             status = $GET (RAB = .rab);
;  890          0881  4             END;
;  891          0882  4         rab [rab$b_rac] = rab$c_seq;
;  892          0883  3         END;
;  893          0884  3 !
;  894          0885  3 ! Get the record from the node file.
;  895          0886  3 !
```

```
;  896          0887  1      IF .status THEN
;  897          0888  1          status = nml$read_node_rec (nmn$c_lis_key_ref,
;  898          0889  1                                      key_value_dsc,
;  899          0890  1                                      rec_node_type,
;  900          0891  1                                      .buffer_dsc, .data_dsc);
;  901          0892  1      !
;  902          0893  1      ! Restore record access mode to keyed in case this is the last time this
;  903          0894  1      ! routine is called for a known record.
;  904          0895  1      !
;  905          0896  1      rab [rab$b_rac] = rab$c_key;
;  906          0897  1      last_RFA0 = .rab [rab$l_rfa0];
;  907          0898  1      last_RFA4 = .rab [rab$w_rfa4];
;  908          0899  1      IF .node_type NEQ .rec_node_type OR
;  909          0900  1         .status EQL rms$_eof OR
;  910          0901  1         .status EQL rms$_rnf THEN
;  911          0902  1          RETURN rms$_eof;
;  912          0903  1      END;
;  913          0904  1  RETURN nml$chkfileio (nma$c_sts_fio,
;  914          0905  1                        .status);
;  915          0906  1  END;              ! Of    nml$read_known_node_rec


                                                .PSECT  $OWN$,NOEXE,2

                                        0022E   .BLKB   2
                                        00230 LAST_RFA0:
                                                .BLKB   4
                                        00234 LAST_RFA4:
                                                .BLKB   2

                                                .EXTRN  SYS$REWIND

                                                .PSECT  $CODE$,NOWRT,2

                                    000C 00000          .ENTRY  NML$READ_KNOWN_NODE_REC, Save R2,R3     ; 0774
                          53 00000000'  00  9E 00002    MOVAB   LAST_RFA4, R3                           ;
                             5E         0C  C2 00009    SUBL2   #12, SP                                 ;
                             52     FE1C  C3  9E 0000C  MOVAB   NML$A_NETNODE_RAB, RAB                  ; 0830
                       04    AE         04  D0 00011    MOVL    #4, KEY_VALUE_DSC                       ; 0831
                       08    AE     04  AC  9E 00015    MOVAB   NODE_TYPE, KEY_VALUE_DSC+4              ; 0832
                             50         01  D0 0001A    MOVL    #1, STATUS                              ; 0833
                             0E     10  AC  E9 0001D    BLBC    REWIND_FLAG, 1$                         ; 0841
                                    FC  A3  D4 00021    CLRL    LAST_RFA0                               ; 0843
                                        63  B4 00024    CLRW    LAST_RFA4                               ; 0844
                                        52  DD 00026    PUSHL   RAB                                     ; 0845
                00000000G   00          01  FB 00028    CALLS   #1, SYS$REWIND                          ;
                             03         50  E8 0002F 1$: BLBS    STATUS, 2$                              ; 0847
                                    0084 31  00032    BRW     9$                                      ;
                             3F     10  AC  E8 00035 2$: BLBS    REWIND_FLAG, 6$                         ; 0853
                             51     FC  A3  D0 00039    MOVL    LAST_RFA0, R1                           ; 0868
                       10    A2         51  D1 0003D    CMPL    R1, T6(RAB)                             ;
                                        06  12 00041    BNEQ    3$                                      ;
                       14    A2         63  B1 00043    CMPW    LAST_RFA4, 20(RAB)                      ; 0869
                                        2C  13 00047    BEQL    5$                                      ;
                                    10  A2  D5 00049 3$: TSTL    16(RAB)                                 ; 0871
                                        05  12 0004C    BNEQ    4$                                      ;
```

```
                                  14   A2  B5 0004E          TSTW    20(RAB)                                 ; 0872
                                       22  13 00051          BEQL    5$
                            1E    A2       02  90 00053 4$:  MOVB    #2, 30(RAB)                             ; 0875
                            10    A2       51  D0 00057      MOVL    R1, 16(RAB)                             ; 0876
                            14    A2       63  B0 0005B      MOVW    LAST_RFA4, 20(RAB)                      ; 0877
                                  51   08  AC  D0 0005F      MOVL    BUFFER_DSC, R1                          ; 0878
                            20    A2       61  B0 00063      MOVW    (R1), 32(RAB)
                            24    A2   04  A1  D0 00067      MOVL    4(R1), 36(RAB)                          ; 0879
                                       52  DD 0006C          PUSHL   RAB                                     ; 0880
                 00000000G  00           01  FB 0006E      CALLS   #1, SYS$GET
                                  1E   A2  94  00075 5$:     CLRB    30(RAB)                                 ; 0882
                                  11       50  E9 00078 6$:  BLBC    STATUS, 7$                              ; 0887
                                  7E   08  AC  7D 0007B      MOVQ    BUFFER_DSC, -(SP)                       ; 0891
                                       08  AE  9F 0007F      PUSHAB  REC_NODE_TYPE                           ; 0888
                                       10  AE  9F 00082      PUSHAB  KEY_VALUE_DSC
                                       03  DD 00085          PUSHL   #3
                       FBE9    CF          05  FB 00087      CALLS   #5, NML$READ_NODE_REC
                         1E    A2          01  90 0008C 7$:  MOVB    #1, 30(RAB)                             ; 0896
                         FC    A3   10  A2  D0 00090      MOVL    16(RAB), LAST_RFA0                      ; 0897
                                  63   14  A2  B0 00095      MOVW    20(RAB), LAST_RFA4                      ; 0898
                                  6E   04  AC  D1 00099      CMPL    NODE_TYPE, REC_NODE_TYPE                ; 0899
                                       12  12 0009D          BNEQ    8$
                 0001827A    8F          50  D1 0009F      CMPL    STATUS, #98938                          ; 0900
                                       09  13 000A6          BEQL    8$
                 000182B2    8F          50  D1 000AB      CMPL    STATUS, #98994                          ; 0901
                                       08  12 000AF          BNEQ    9$
                                  50 0001827A  8F  D0 000B1 8$:  MOVL    #98938, R0                          ; 0902
                                                   04 000B8  RET
                                       50  DD 000B9 9$:     PUSHL   STATUS                                  ; 0905
                                  7E       12  CE 000BB      MNEGL   #18, -(SP)                              ; 0904
                 00000000G  00           02  FB 000BE      CALLS   #2, NML$CHKFILEIO
                                                   04 000C5  RET                                             ; 0906

; Routine Size:  198 bytes,    Routine Base:  $CODE$ + 0499
```

```
;   917          0907  1 %SBTTL 'nml$create_node_db                Create node permanent database file'
;   918          0908  1 GLOBAL ROUTINE nml$create_node_db (file_name_dsc, fab) =
;   919          0909  1
;   920          0910  1 !++
;   921          0911  1 ! FUNCTIONAL DESCRIPTION:
;   922          0912  1 !       This routine is called to create a new node database file under two
;   923          0913  1 !       conditions:
;   924          0914  1 !             - None already exists.
;   925          0915  1 !             - If the node permanent database has only 1 key - it's the
;   926          0916  1 !               old node database format, and must be converted to four
;   927          0917  1 !               keys (this conversion is for performance reasons).  Create
;   928          0918  1 !               the file here, convert it later.
;   929          0919  1 !
;   930          0920  1 ! FORMAL PARAMETERS:
;   931          0921  1 !       FILE_NAME_DSC   Descriptor of name of file.  Used because, when
;   932          0922  1 !                       converting from the old database format to the new,
;   933          0923  1 !                       the new file is given a temporary file name until
;   934          0924  1 !                       complete.
;   935          0925  1 !       FAB             Address at which to return address of FAB.
;   936          0926  1 !
;   937          0927  1 ! ROUTINE VALUE:
;   938          0928  1 ! COMPLETION CODES:
;   939          0929  1 !
;   940          0930  1 !       Failure or RMS error
;   941          0931  1 !
;   942          0932  1 !--
;   943          0933  1
;   944          0934  2 BEGIN
;   945          0935  2
;   946          0936  2 MAP
;   947          0937  2     file_name_dsc: REF VECTOR;
;   948          0938  2
;   949          0939  2 LOCAL
;   950          0940  2     status;
;   951          0941  2
;   952          0942  2 .fab = nml$a_netnode_fab;
;   953        P 0943  2 $FAB_INIT (       FAB = nml$a_netnode_fab,
;   954        P 0944  2                   ALQ = 60,                       ! Initial file block size.
;   955        P 0945  2                   BKS = 3,                        ! Bucket size
;   956        P 0946  2                   FAC = (UPD, PUT, GET, DEL),     ! File access options
;   957        P 0947  2                   DNM = 'SYS$SYSTEM:.DAT',        ! Default filename string
;   958        P 0948  2                   FNA = .file_name_dsc [1],       ! File name
;   959        P 0949  2                   FNS = .file_name_dsc [0],       ! File name size
;   960        P 0950  2                   FOP = (CBT,MXV),                ! File Options (contiguous best
;   961        P 0951  2                                                   !       try, max versions)
;   962        P 0952  2                   ORG = IDX,                      ! Organization = indexed
;   963        P 0953  2                   RFM = VAR,                      ! Record format = variable
;   964        P 0954  2                   SHR = (UPD, PUT, GET, DEL),     ! File sharing options
;   965          0955  2                   XAB = nml$a_node_address_xab);  ! XAB Chain
;   966          0956  2 !
;   967          0957  2 ! Set up the XABs to describe the four keys which will be used
;   968          0958  2 ! to get information from the file.
;   969          0959  2 !
;   970          0960  2 !
;   971          0961  2 ! First, initialize primary key XAB with key = node address.  Allow duplicates
;   972          0962  2 ! for this key because any loopnode can have an address of zero.
;   973          0963  2 !
```

```
;   974        P 0964  2 $XABKEY_INIT (XAB = nml$a_node_address_xab,      ! XAB address
;   975        P 0965  2                 DTP = BN2,                      ! Key data type = 2 byte binary
;   976        P 0966  2                 FLG = (DUP, DAT_NCMPR, IDX_NCMPR,
;   977        P 0967  2                        KEY_NCMPR),              ! Key flags
;   978        P 0968  2                 KREF = nmn$c_add_key_ref,       ! Key reference number
;   979        P 0969  2                 POS = 0,                        ! Key position in record
;   980        P 0970  2                 SIZ = nmn$c_add_key_len,        ! Key size in record
;   981           0971  2                 NXT = nml$a_node_type_xab);     ! XAB chain pointer
;   982           0972  2 !
;   983           0973  2 ! Next, initialize key XAB with key = node type (executor, remote, loop).
;   984           0974  2 !
;   985        P 0975  2 $XABKEY_INIT (XAB = nml$a_node_type_xab,        ! XAB address
;   986        P 0976  2                 DTP = BN2,                      ! Key data type = 2 byte binary
;   987        P 0977  2                 FLG = (CHG, DUP, IDX_NCMPR),    ! Key flags
;   988        P 0978  2                 KREF = nmn$c_typ_key_ref,       ! Key reference number
;   989        P 0979  2                 POS = 2,                        ! Key position in record
;   990        P 0980  2                 SIZ = nmn$c_typ_key_len,        ! Key size in record
;   991           0981  2                 NXT = nml$a_node_name_xab);     ! XAB chain
;   992           0982  2
;   993           0983  2 !
;   994           0984  2 ! Initialize key XAB with key = node name
;   995           0985  2 !
;   996        P 0986  2 $XABKEY_INIT (XAB = nml$a_node_name_xab,        ! XAB address
;   997        P 0987  2                 DTP = STG,                      ! Key data type = string
;   998        P 0988  2                 FLG = (CHG, NUL, IDX_NCMPR),    ! Key flags
;   999        P 0989  2                 KREF = nmn$c_nam_key_ref,       ! Key reference number
;  1000        P 0990  2                 POS = 4,                        ! Key position in record
;  1001        P 0991  2                 SIZ = nmn$c_nam_key_len,        ! Key size in record
;  1002        P 0992  2                 NUL = %C' ',                    ! Null key = blank
;  1003           0993  2                 NXT = nml$a_node_list_xab);     ! XAB chain
;  1004           0994  2
;  1005           0995  2 !
;  1006           0996  2 ! Initialize key XAB with key = list node.
;  1007           0997  2 ! This key concatenates the the node address key with the node type key to
;  1008           0998  2 ! allow the LIST command to get nodes by type and, within type, sequentially
;  1009           0999  2 ! by address.  The list key value must be set up with a zero for the node
;  1010           1000  2 ! address; hence when you do a $GET of (type OR 0) with a match type of GTR,
;  1011           1001  2 ! it will get the first node of that type in the file.  Subsequent sequential
;  1012           1002  2 ! reads will return the nodes of that type in ascending order by address.
;  1013           1003  2 !
;  1014        P 1004  2 $XABKEY_INIT (XAB = nml$a_node_list_xab,        ! XAB address
;  1015        P 1005  2                 DTP = BN4,                      ! Key data type = 4 byte binary
;  1016        P 1006  2                 FLG = (CHG, DUP, IDX_NCMPR),    ! Key flags
;  1017        P 1007  2                 KREF = nmn$c_lis_key_ref,       ! Key reference number
;  1018        P 1008  2                 POS = 0,                        ! Key position in record
;  1019        P 1009  2                 SIZ = nmn$c_lis_key_len,        ! Key size in record
;  1020           1010  2                 NXT = nml$a_protection_xab);    ! XAB chain
;  1021           1011  2
;  1022        P 1012  2 $XABPRO_INIT (XAB = nml$a_protection_xab,       ! XAB address
;  1023        P 1013  2                 UIC = (1, 4),                   ! UIC of owner
;  1024           1014  2                 PRO = (RWED, RWED, , ));        ! Protection (group and world
;  1025           1015  2                                                 !       no access)
;  1026           1016  2
;  1027           1017  2
;  1028           1018  2 status = $CREATE (FAB = nml$a_netnode_fab);
;  1029           1019  2
;  1030           1020  2 IF .status THEN
```

```
;  1031          1021  2     nml$logfileop (dbg$c_fileio,
;  1032          1022  2                    nma$c_opn_node,
;  1033          1023  2                    $ASCID ('file created'));
;  1034          1024  2 RETURN .status;
;  1035          1025  2
;  1036          1026  1 END;             ! of            nml$create_node_db


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

54  41  44  2E  3A  4D  45  54  53  59  53  24  53  59  53  00080 P.AAT:  .ASCII  \SYS$SYSTEM:.DAT\                              ;
            64  65  74  61  65  72  63  20  65  6C  69  66  00088 P.AAV:  .ASCII  \file created\                                 ;
                                                            000CB       .BLKB   1
                                                    0000000C 000CC P.AAU:  .LONG   12                                          ;
                                                    00000000' 000D0        .ADDRESS P.AAV                                      ;

                                                                $RMS_PTR=       NML$A_NETNODE_FAB
                                                                $RMS_PTR=       NML$A_NODE_ADDRESS_XAB
                                                                $RMS_PTR=       NML$A_NODE_TYPE_XAB
                                                                $RMS_PTR=       NML$A_NODE_NAME_XAB
                                                                $RMS_PTR=       NML$A_NODE_LIST_XAB
                                                                $RMS_PTR=       NML$A_PROTECTION_XAB
                                                                .EXTRN  SYS$CREATE

                                                                .PSECT  $CODE$,NOWRT,2

                                                        007C 00000      .ENTRY  NML$CREATE_NODE_DB, Save R2,R3,R4,R5,R6      ; 0908
                                    56 00000000'    00   9E 00002      MOVAB   NML$A_NETNODE_FAB, R6                        ;
                                08  BC              66   9E 00009      MOVAB   NML$A_NETNODE_FAB, @FAB                      ; 0942
    0050    8F          00          6E              00   2C 0000D      MOVC5   #0, (SP), #0, #80, $RMS_PTR                  ; 0955
                                                    66       00014      ;                                                    ;
                                66      5003        8F   B0 00015      MOVW    #20483, $RMS_PTR                             ;
                              04    A6  00200002      8F   D0 0001A      MOVL    #2097154, $RMS_PTR+4                         ;
                              10    A6                3C   D0 00022      MOVL    #60, $RMS_PTR+T6                             ;
                              16    A6      0F0F      8F   B0 00026      MOVW    #3855, $RMS_PTR+22                           ;
                              1D    A6                20   90 0002C      MOVB    #32, $RMS_PTR+29                             ;
                              1F    A6                02   90 00030      MOVB    #2, $RMS_PTR+31                              ;
                              24    A6      00F8      C6   9E 00034      MOVAB   NML$A_NODE_ADDRESS_XAB, $RMS_PTR+36           ;
                                    50        04      AC   D0 0003A      MOVL    FILE_NAME_DSC, R0                            ;
                              2C    A6        04      A0   D0 0003E      MOVL    4(R0), $RMS_PTR+44                           ;
                              30    A6  00000000'     00   9E 00043      MOVAB   P.AAT, $RMS_PTR+48                           ;
                              34    A6                60   90 0004B      MOVB    (R0), $RMS_PTR+52                            ;
                              35    A6                0F   90 0004F      MOVB    #15, $RMS_PTR+53                             ;
                              3E    A6                03   90 00053      MOVB    #3, $RMS_PTR+62                              ;
    004C    8F          00          6E              00   2C 00057      MOVC5   #0, (SP), #0, #76, $RMS_PTR                  ; 0971
                                                    00F8 C6      0005E      ;                                                    ;
                                00F8    C6      4C15    8F   B0 00061      MOVW    #19477, $RMS_PTR                             ;
                                00FC    C6      0190    C6   9E 00068      MOVAB   NML$A_NODE_TYPE_XAB, $RMS_PTR+4              ;
                                010A    C6      02C9    8F   B0 0006F      MOVW    #713, $RMS_PTR+T8                            ;
                                                010F    C6   94 00076      CLRB    $RMS_PTR+23                                  ;
                                0126    C6              02   90 0007A      MOVB    #2, $RMS_PTR+46                              ;
    004C    8F          00          6E              00   2C 0007F      MOVC5   #0, (SP), #0, #76, $RMS_PTR                  ; 0981
                                                    0190 C6      00086      ;                                                    ;
                                0190    C6      4C15    8F   B0 00089      MOVW    #19477, $RMS_PTR                             ;
                                0194    C6      0144    C6   9E 00090      MOVAB   NML$A_NODE_NAME_XAB, $RMS_PTR+4              ;
                                01A2    C6      020B    8F   B0 00097      MOVW    #523, $RMS_PTR+T8                            ;
```

```
                         01A7   C6              01  90 0009E          MOVB    #1, $RMS_PTR+23                 ;
                         01AE   C6              02  B0 000A3          MOVW    #2, $RMS_PTR+30                 ;
                         01BE   C6              02  90 000A8          MOVB    #2, $RMS_PTR+46                 ;
    004C   8F      00           6E              00  2C 000AD          MOVC5   #0, (SP), #0, #76, $RMS_PTR     ; 0993
                                        0144    C6     000B4
                         0144   C6      4C15    8F  B0 000B7          MOVW    #19477, $RMS_PTR
                         0148   C6      01DC    C6  9E 000BE          MOVAB   NML$A_NODE_LIST_XAB, $RMS_PTR+4
                         0156   C6              0E  B0 000C5          MOVW    #14, $RMS_PTR+18
                         0159   C6              20  90 000CA          MOVB    #32, $RMS_PTR+21
                         015B   C6              02  90 000CF          MOVB    #2, $RMS_PTR+23
                         0162   C6              04  B0 000D4          MOVW    #4, $RMS_PTR+30
                         0172   C6              06  90 000D9          MOVB    #6, $RMS_PTR+46
    004C   8F      00           6E              00  2C 000DE          MOVC5   #0, (SP), #0, #76, $RMS_PTR     ; 1010
                                        01DC    C6     000E5
                         01DC   C6      4C15    8F  B0 000E8          MOVW    #19477, $RMS_PTR
                         01E0   C6      0094    C6  9E 000EF          MOVAB   NML$A_PROTECTION_XAB, $RMS_PTR+4
                         01EE   C6      040B    8F  B0 000F6          MOVW    #1035, $RMS_PTR+18
                         01F3   C6              03  90 000FD          MOVB    #3, $RMS_PTR+23
                         020A   C6              04  90 00102          MOVB    #4, $RMS_PTR+46
    0058   8F      00           6E              00  2C 00107          MOVC5   #0, (SP), #0, #88, $RMS_PTR     ; 1014
                                        0094    C6     0010E
                         0094   C6      5813    8F  B0 00111          MOVW    #22547, $RMS_PTR
                         009C   C6      FF00    8F  B0 00118          MOVW    #-256, $RMS_PTR+8
                         00A0   C6  00010004    8F  D0 0011F          MOVL    #65540, $RMS_PTR+12
                                                56  DD 00128          PUSHL   R6                              ; 1018
                00000000G        00            01  FB 0012A          CALLS   #1, SYS$CREATE
                                 52            50  D0 00131          MOVL    R0, STATUS
                                 10            52  E9 00134          BLBC    STATUS, 1$                      ; 1020
                                       00000000' 00  9F 00137          PUSHAB  P.AAU                           ; 1023
                                 7E            01  7D 0013D          MOVQ    #1, -(SP)                       ; 1021
                00000000G        00            03  FB 00140          CALLS   #3, NML$LOGFILEOP
                                 50            52  D0 00147 1$:      MOVL    STATUS, R0                      ; 1024
                                               04  00014A          RET                                     ; 1026

; Routine Size:  331 bytes,    Routine Base:  $CODE$ + 055F
```

```
;  1038          1027  1 %SBTTL 'nml$connect_node_rab    Open node permanent database file'
;  1039          1028  1 GLOBAL ROUTINE nml$connect_node_rab  =
;  1040          1029  1
;  1041          1030  1 !++
;  1042          1031  1 ! FUNCTIONAL DESCRIPTION:
;  1043          1032  1 !       This builds a RAB for accessing the node database file and
;  1044          1033  1 !       issues a connect.
;  1045          1034  1 !
;  1046          1035  1 ! FORMAL PARAMETERS:
;  1047          1036  1 !       NONE
;  1048          1037  1 !
;  1049          1038  1 ! ROUTINE VALUE:
;  1050          1039  1 ! COMPLETION CODES:
;  1051          1040  1 !       Failure or RMS error
;  1052          1041  1 !
;  1053          1042  1 !--
;  1054          1043  1
;  1055          1044  2 BEGIN
;  1056          1045  2
;  1057          1046  2 !
;  1058          1047  2 ! Initialize most of RAB here.  Init it to use the primary key
;  1059          1048  2 ! (node address) to begin with.  This is changed when other keys
;  1060          1049  2 ! are needed.
;  1061          1050  2 !
;  1062        P 1051  2 $RAB_INIT (RAB = nml$a_netnode_rab,
;  1063        P 1052  2               FAB = nml$a_netnode_fab,
;  1064        P 1053  2               KRF = nmn$c_add_key_ref,     ! primary key = node address
;  1065        P 1054  2               MBF = 10,
;  1066        P 1055  2               RAC = KEY,
;  1067          1056  2               ROP = UIF);
;  1068          1057  2
;  1069          1058  2 !
;  1070          1059  2 ! Connect RMS record stream.
;  1071          1060  2 !
;  1072          1061  2 RETURN $CONNECT (RAB = nml$a_netnode_rab);
;  1073          1062  1 END;                ! of   nml$connect_node_rab


                                                $RMS_PTR=           NML$A_NETNODE_RAB
                                                        .EXTRN  SYS$CONNECT

                                                 007C 00000         .ENTRY  NML$CONNECT_NODE_RAB, Save R2,R3,R4,R5,R6   ; 1028
                                56 00000000'  00  9E 00002         MOVAB   $RMS_PTR, R6
    0044   8F           00         6E          00  2C 00009         MOVC5   #0, (SP), #0, #68, $RMS_PTR                 ; 1056
                                                66       00010
                                66     4401    8F  B0 00011         MOVW    #17409, $RMS_PTR
                          04    A6             10  D0 00016         MOVL    #16, $RMS_PTR+4
                          1E    A6             01  90 0001A         MOVB    #1, $RMS_PTR+30
                          35    A6     0A00    8F  B0 0001E         MOVW    #2560, $RMS_PTR+53
                          3C    A6       B0    A6  9E 00024         MOVAB   NML$A_NETNODE_FAB, $RMS_PTR+60
                                                56  DD 00029         PUSHL   R6                                          ; 1061
                    00000000G   00             01  FB 0002B         CALLS   #1, SYS$CONNECT
                                                04  00032         RET                                                 ; 1062

; Routine Size:  51 bytes,    Routine Base:  $CODE$ + 06AA
```

```
; 1074          1063  1
; 1075          1064  1 END                                   ! End of module
; 1076          1065  1
; 1077          1066  0 ELUDOM
```

```
;                                PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  $OWN$                          566  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                         212  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $GLOBAL$                         8  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        1757  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
```

```
;                            Library Statistics

;                                    -------- Symbols --------      Pages       Processing
;       File                         Total    Loaded   Percent      Mapped      Time

;  _$255$DUA28:[NML.OBJ]NMLLIB.L32;1     341        45        13          27        00:00.1
;  _$255$DUA28:[SHRLIB]NMALIBRY.L32;1    887         6         0          47        00:00.2
;  _$255$DUA28:[SYSLIB]STARLET.L32;1    9776       151         1         581        00:02.1
```

```
;                                COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:NMLNODFIL/OBJ=OBJ$:NMLNODFIL MSRC$:NMLNODFIL/UPDATE=(ENH$:NMLNODFIL)

; Size:          1757 code + 786 data bytes
; Run Time:        00:40.9
; Elapsed Time:    01:25.8
; Lines/CPU Min:    1565
; Lexemes/CPU-Min: 32825
; Memory Used:  219 pages
; Compilation Complete
```

0285 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

NMLPURGE
LIS

NMLPARINI
LIS

NMLNODFIL
LIS

NMLREAD
LIS

NMLPARPRM
LIS

NMLPMANIP
LIS